

DEVICE AND METHOD FOR PROCESSING DATA, DEVICE AND METHOD FOR REPRODUCING DATA, AND RECORDING MEDIUM

Publication number: JP2001024985

Publication date: 2001-01-26

Inventor: KATO MOTOKI; HAMADA TOSHIYA; NAKAMURA MASANOBU

Applicant: SONY CORP

Classification:

- international: H04N5/92; G11B20/10; H04N7/24; H04N7/26;
H04N5/92; G11B20/10; H04N7/24; H04N7/26; (IPC1-7):
H04N5/92; G11B20/10; H04N7/24

- European:

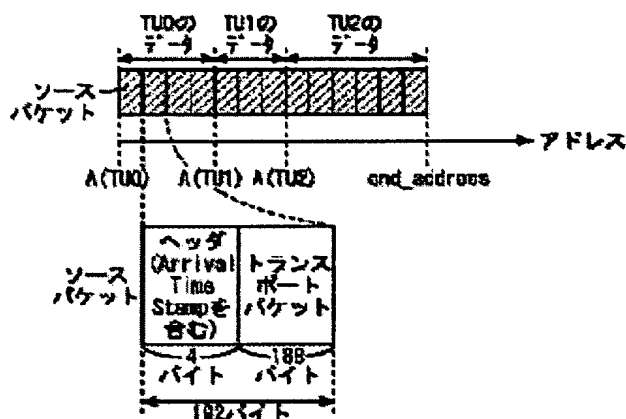
Application number: JP20000075128; 20000317

Priority number(s): JP20000075128; 20000317; JP19990126164; 19990506

Report a data error here

Abstract of JP2001024985

PROBLEM TO BE SOLVED: To efficiently record an encoded stream onto a recording medium and to speedily perform random access. **SOLUTION:** An inputted transport stream is divided for each time unit having a prescribed length and among packets included in each time unit, the packets of a channel, to which recording is designated, are collected and recorded on the recording medium without adding any dummy packet. The data length of a time unit TU0 expressed with a value subtracting a leading address A (TU0) of the time unit TU0 from a leading address 'time unit address' of a recorded packet among respective time units TU0, TU1 and TU2 such as leading address A (TU1) of a time unit TU1, for example, is correspondently recorded on the recording medium as a time unit map.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-24985

(P2001-24985A)

(43) 公開日 平成13年1月26日 (2001.1.26)

(51) Int.Cl. ⁷	識別記号	F I	テマコード*(参考)
H 0 4 N 5/92		H 0 4 N 5/92	H 5 C 0 5 3
G 1 1 B 20/10	3 0 1	G 1 1 B 20/10	3 0 1 Z 5 C 0 5 9
H 0 4 N 7/24		H 0 4 N 7/13	Z 5 D 0 4 4

審査請求 未請求 請求項の数26 O L (全 42 頁)

(21) 出願番号 特願2000-75128(P2000-75128)
(22) 出願日 平成12年3月17日 (2000.3.17)
(31) 優先権主張番号 特願平11-126164
(32) 優先日 平成11年5月6日 (1999.5.6)
(33) 優先権主張国 日本 (J P)

(71) 出願人 000002185
ソニー株式会社
東京都品川区北品川6丁目7番35号
(72) 発明者 加藤 元樹
東京都品川区北品川6丁目7番35号 ソニ
ー株式会社内
(72) 発明者 浜田 俊也
東京都品川区北品川6丁目7番35号 ソニ
ー株式会社内
(74) 代理人 100082131
弁理士 稲本 義雄

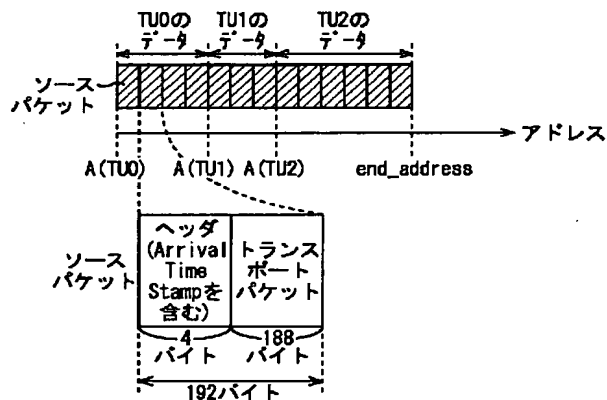
最終頁に続く

(54) 【発明の名称】 データ処理装置および方法、データ再生装置および方法、並びに記録媒体

(57) 【要約】

【課題】 記録媒体に対して、符号化ストリームを効率的に記録し、迅速にランダムアクセスできるようにする。

【解決手段】 入力されたトランスポートストリームが、所定の長さのタイムユニット毎に区分され、各タイムユニットに含まれるパケットの内、記録が指定されたチャネルのパケットが集められ、ダミーパケットが付加されることなく、記録媒体に記録される。各タイムユニットTU0、TU1、TU2のパケットの内、記録されたパケットの先頭のアドレスtime unit addressと、例えば、タイムユニットTU1の先頭のアドレスA(TU1)から、タイムユニットTU0の先頭のアドレスA(TU0)を減算した値で表わされるタイムユニットTU0のデータ長が、対応され、タイムユニットマップとして記録媒体に記録される。



【特許請求の範囲】

【請求項1】 入力された符号化ストリームに含まれるデータを処理するデータ処理装置において、
入力された前記符号化ストリームを、所定のタイムユニット毎に区分する区分手段と、
前記区分手段により区分された前記符号化ストリームのタイムユニット毎のデータのアドレスを示すタイムユニットマップを作成する第1の作成手段とを備えることを特徴とするデータ処理装置。

【請求項2】 前記タイムユニットマップは、前記タイムユニット毎のデータのアドレスとして、タイムユニット毎の先頭のパケットのパケット番号を保持することを特徴とする請求項1に記載のデータ処理装置。

【請求項3】 前記タイムユニットマップは、前記タイムユニットの先頭のデータのアドレスと、そのタイムユニット内に含まれるデータ量に対応するアドレスの間隔とを対応して保持することを特徴とする請求項1に記載のデータ処理装置。

【請求項4】 前記符号化ストリームとともに、前記タイムユニットマップをファイル化するファイル化手段をさらに備えることを特徴とする請求項1に記載のデータ処理装置。

【請求項5】 前記ファイル化手段によりファイル化されたデータを記録媒体に記録する記録手段をさらに備えることを特徴とする請求項4に記載のデータ処理装置。

【請求項6】 前記第1の作成手段は、前記符号化ストリームが編集されたとき、前記タイムユニットマップを変更することを特徴とする請求項1に記載のデータ処理装置。

【請求項7】 前記符号化ストリームのエントリポイントの位置を示す、前記タイムユニットマップに従属するエントリポイントマップを作成する第2の作成手段をさらに備えることを特徴とする請求項1に記載のデータ処理装置。

【請求項8】 前記符号化ストリームが、複数のプログラムを多重化して生成された符号化ストリームである場合、前記第2の作成手段は、前記エントリポイントマップを、前記プログラム毎に作成することを特徴とする請求項7に記載のデータ処理装置。

【請求項9】 前記第2の作成手段は、前記符号化ストリームが編集されたとき、前記エントリポイントマップを変更することを特徴とする請求項7に記載のデータ処理装置。

【請求項10】 入力された符号化ストリームに含まれるデータを処理するデータ処理装置のデータ処理方法において、
入力された前記符号化ストリームを、所定のタイムユニット毎に区分する区分ステップと、
前記区分ステップの処理により区分された前記符号化ストリームのタイムユニット毎のデータのアドレスを示す

タイムユニットマップを作成する作成ステップとを含むことを特徴とするデータ処理方法。

【請求項11】 入力された符号化ストリームに含まれるデータを処理するプログラムであって、
入力された前記符号化ストリームを、所定のタイムユニット毎に区分する区分ステップと、
前記区分ステップの処理により区分された前記符号化ストリームのタイムユニット毎のデータのアドレスを示すタイムユニットマップを作成する作成ステップとを含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項12】 記録媒体に記録されている、符号化ストリームのタイムユニット毎のデータのアドレスを示すタイムユニットマップを再生する第1の再生手段と、
前記第1の再生手段により再生された、前記タイムユニットマップに基づいて、前記記録媒体に記録されている前記符号化ストリームを、任意の位置から再生する第2の再生手段とを備えることを特徴とするデータ再生装置。

【請求項13】 前記第1の再生手段は、前記符号化ストリームのエントリポイントの位置を示す、前記タイムユニットマップに従属するエントリポイントマップもさらに再生し、
前記第2の再生手段は、前記第1の再生手段により再生された、前記タイムユニットマップとエントリポイントマップに基づいて、前記記録媒体に記録されている前記符号化ストリームを、任意の位置から再生することとを特徴とする請求項12に記載のデータ再生装置。

【請求項14】 記録媒体に記録されている、符号化ストリームのタイムユニット毎のデータのアドレスを示すタイムユニットマップを再生する第1の再生ステップと、
前記第1の再生ステップの処理により再生された、前記タイムユニットマップに基づいて、前記記録媒体に記録されている前記符号化ストリームを、任意の位置から再生する第2の再生ステップとを含むことを特徴とするデータ再生方法。

【請求項15】 記録媒体に記録されている、符号化ストリームのタイムユニット毎のデータのアドレスを示すタイムユニットマップを再生する第1の再生ステップと、
前記第1の再生ステップの処理により再生された、前記タイムユニットマップに基づいて、前記記録媒体に記録されている前記符号化ストリームを、任意の位置から再生する第2の再生ステップとを含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、データ処理装置お

よび方法、並びに記録媒体に関し、特に符号化ストリームから特徴点情報を抽出することにより、複数のプログラムが多重化されている場合においても、迅速にランダムアクセスができるようにしたデータ処理装置および方法、並びに記録媒体に関する。

【0002】

【従来の技術】ヨーロッパのDVB(Digital Video Broadcast)や日本のデジタルBS放送などの多チャンネルデジタルテレビジョン放送では、MPEG(Moving Picture Experts Group) 2トランスポートストリームが使われる。トランスポートストリームは、トランスポートパケットが連続したストリームであり、トランスポートパケットは、例えば、MPEG 2ビデオストリームやMPEG 1オーディオストリームがパケット化されたものである。放送の電波で伝送される1本のトランスポートストリームには、1つまたは複数のAV(Audio Visual)プログラムが多重化されている。一般に、各チャンネルのAVプログラムは、お互いに独立している。

【0003】したがって、放送で送られるトランスポートストリームを家庭の受信機でそのまま受信し、記録すれば、そのトランスポートストリームのすべてのチャンネルのプログラムを同時に記録することができる。また、放送で送られるトランスポートストリームの中からユーザによって選択された幾つかのチャンネルのAVプログラムのトランスポートストリームを分離したものを記録すれば、選択された任意の数のチャンネルのプログラムを同時に記録することができる。

【0004】図1に従来のトランスポートストリームの記録方法の例を示す。図1(A)は、複数のAVプログラムが多重化されたトランスポートストリームを示す。ここで横軸は、時間であり、 Δt の間隔のタイムユニット TU_i ($i=0, 1, 2, \dots$) 毎に区切られている。入力トランスポートストリームの中から1つまたは複数のAVプログラムが選択される。選択されたトランスポートパケットは斜線を施して示されている。選択されたトランスポートパケットは、一般に、図1(B)に示すように、不規則なタイミングで現れ、タイムユニット TU_i 毎のトランスポートパケットの数は変化する。

【0005】 Δt の間隔のタイムユニット TU_i 毎の選択されたトランスポートパケットは、図2に示すように、間隔を詰めて記録媒体に記録される。この時、各トランスポートパケットは、それぞれのストリーム上の時刻を示すタイムスタンプを付加して記録される。このタイムスタンプは、例えば、DV(Digital Video)フォーマット(Specification Of Consumer-Use Digital VCRs using 6.3mm magnetic tape (HD digital VCR conference), PART 7 DVB Specifications of Consumer-Use Digital VCR)で規定されているところのトランスポートパケットに付加される4バイト長のTSP_extra_headerと同様のものである。

【0006】図2において、横軸は記録されたトランスポートストリームのバイト位置を示すアドレスである。図1(B)に示すような可変ビットレートのトランスポートストリームが入力されると、記録装置は、図2に示すように、ダミーデータを入れて、固定の記録レートでデータを記録する。したがって、記録されたトランスポートストリームの時間の経過に対するデータ量は比例する。すなわち、タイムユニットあたりの記録データ量を x とすると、 n 番目 ($n=0, 1, 2, \dots$) のタイムユニットの先頭データのバイト位置は、 n 倍の x となる。

【0007】

【発明が解決しようとする課題】このように、従来の記録方法は、ダミーデータを挿入して一定の記録レートにしているため、トランスポートストリームの記録効率が良くない。しかしながら、ダミーデータを挿入しないと、記録されたトランスポートストリームの時間の経過とファイルのデータ量が比例しなくなるので、トランスポートストリームの時間軸上の所定の位置のデータにアクセスする場合、データのアクセス性が悪くなる問題が発生する。

【0008】また、一般に、MPEG 2ビデオのストリームでは、0.5秒くらいの間隔でIピクチャが符号化され、それ以外のピクチャはPピクチャまたはBピクチャとして符号化される。したがって、MPEG 2ビデオのストリームが記録された記録媒体から、ビデオ信号を高速再生する場合は、Iピクチャをサーチしなければならない。ところが、デジタル放送等のトランスポートストリームが記録された記録媒体から、ランダムアクセスにより再生を行う場合、Iピクチャの開始バイトを効率よくサーチすることが難しかった。すなわち、記録媒体上のトランスポートストリームのランダムなバイト位置から読み出したビデオストリームのシンタックスを解析し、Iピクチャやオーディオフレームの開始バイトがサーチされていた。そのため、場合によっては、Iピクチャのサーチに時間がかかってしまい、ユーザ入力に対して応答の速いランダムアクセス再生をすることが困難であった。

【0009】本発明は、このような状況に鑑みてなされたものであり、複数のプログラムが多重化されている場合においても、迅速に、ランダムアクセスができるようにするものである。

【0010】

【課題を解決するための手段】本発明の第1のデータ処理装置は、入力された符号化ストリームを、所定のタイムユニット毎に区分する区分手段と、区分手段により区分された符号化ストリームのタイムユニット毎のデータのアドレスを示すタイムユニットマップを作成する第1の作成手段とを備えることを特徴とする。

【0011】前記タイムユニットマップは、タイムユニット毎のデータのアドレスとして、タイムユニット毎の先頭のパケットのパケット番号を保持するようにするこ

とができる。

【0012】前記タイムユニットマップは、タイムユニットの先頭のデータのアドレスと、そのタイムユニット内に含まれるデータ量に対応するアドレスの間隔とを対応して保持するようにすることができる。

【0013】前記符号化ストリームとともに、タイムユニットマップをファイル化するファイル化手段をさらに備えるようにすることができる。

【0014】前記ファイル化手段によりファイル化されたデータを記録媒体に記録する記録手段をさらに備えるようにすることができる。

【0015】前記第1の作成手段は、符号化ストリームが編集されたとき、タイムユニットマップを変更するようにすることができる。

【0016】前記符号化ストリームのエン트리ポイントの位置を示す、タイムユニットマップに従属するエン트리ポイントマップを作成する第2の作成手段をさらに備えるようにすることができる。

【0017】前記符号化ストリームが、複数のプログラムを多重化して生成された符号化ストリームである場合、第2の作成手段は、エン트리ポイントマップを、プログラム毎に作成するようにすることができる。

【0018】前記第2の作成手段は、符号化ストリームが編集されたとき、エン트리ポイントマップを変更するようにすることができる。

【0019】本発明第1のデータ処理方法は、入力された符号化ストリームを、所定のタイムユニット毎に区分する区分ステップと、区分ステップの処理により区分された符号化ストリームのタイムユニット毎のデータのアドレスを示すタイムユニットマップを作成する作成ステップとを含むことを特徴とする。

【0020】本発明第1の記録媒体のプログラムは、入力された符号化ストリームを、所定のタイムユニット毎に区分する区分ステップと、区分ステップの処理により区分された符号化ストリームのタイムユニット毎のデータのアドレスを示すタイムユニットマップを作成する作成ステップとを含むことを特徴とする。

【0021】本発明第2のデータ再生装置は、記録媒体に記録されている、符号化ストリームのタイムユニット毎のデータのアドレスを示すタイムユニットマップを再生する第1の再生手段と、第1の再生手段により再生された、タイムユニットマップに基づいて、記録媒体に記録されている符号化ストリームを、任意の位置から再生する第2の再生手段とを備えることを特徴とする。

【0022】第1の再生手段は、符号化ストリームのエン트리ポイントの位置を示す、タイムユニットマップに従属するエン트리ポイントマップもさらに再生し、第2の再生手段は、第1の再生手段により再生された、タイムユニットマップとエン트리ポイントマップに基づいて、記録媒体に記録されている符号化ストリームを、任

意の位置から再生するようにすることができる。

【0023】本発明第2のデータ再生方法は、記録媒体に記録されている、符号化ストリームのタイムユニット毎のデータのアドレスを示すタイムユニットマップを再生する第1の再生ステップと、第1の再生ステップの処理により再生された、タイムユニットマップに基づいて、記録媒体に記録されている符号化ストリームを、任意の位置から再生する第2の再生ステップとを含むことを特徴とする。

【0024】本発明第2の記録媒体のプログラムは、記録媒体に記録されている、符号化ストリームのタイムユニット毎のデータのアドレスを示すタイムユニットマップを再生する第1の再生ステップと、第1の再生ステップの処理により再生された、タイムユニットマップに基づいて、記録媒体に記録されている符号化ストリームを、任意の位置から再生する第2の再生ステップとを含むことを特徴とする。本発明第1のデータ処理装置、データ処理方法および、記録媒体のプログラムにおいては、入力された符号化ストリームが、タイムユニット毎に区分され、そのタイムユニット毎のデータのアドレスを示すタイムユニットマップと、符号化ストリームのエン트리ポイントの位置を示すタイムユニットマップに従属するエン트리ポイントマップが作成される。

【0025】本発明の第2のデータ処理装置、データ処理方法および、記録媒体のプログラムにおいては、記録媒体から符号化ストリームのタイムユニット毎のデータのアドレスを示すタイムユニットマップと、符号化ストリームのエン트리ポイントの位置を示すタイムユニットマップに従属するエン트리ポイントマップが再生され、それらに基づいて符号化ストリームが任意の位置から再生される。

【0026】

【発明の実施の形態】以下、符号化ストリームが1つまたは複数のプログラムが多重化されている多重化ストリームである場合を例として本発明を説明するが、本発明は、符号化ストリームがMPEGビデオストリーム等のエレメンタリーストリームであっても適用できるものである。

【0027】最初に、本発明の基本的な原理について説明する。本発明の動画記録装置は、1つまたは複数のプログラムが多重化されているトランスポートストリームをディスク、テープなどの記録媒体に記録する時に、ストリーム上の時間を所定のタイムユニット（単位時間）毎に区切り、タイムユニット毎のデータのストリーム上のアドレスを計算する。そして、このタイムユニット毎のデータのストリーム上のアドレスを示すタイムユニットマップが作成される。さらに、記録するトランスポートストリームのプログラム毎のエン트리ポイント（ランダムアクセスポイント）の場所を示すエン트리ポイントマップが作成される。エン트리ポイントマッ

プは、タイムユニットマップに従属する構造を有する。このタイムユニットマップについて、以下に説明する。

【0028】図3は、複数のAVプログラムが多重化されたトランスポートストリームを示す。ここで横軸は、時間を示し、 Δt の間隔のタイムユニットTU i ($i=0, 1, 2, \dots$) 毎に区切られている。文字TUの後ろに続く数字 i は、タイムユニットTUの時間順序を示す。最初のオリジナル記録の時は、すべてのタイムユニットTUの時間長は同じ値 Δt である。値 Δt の大きさは、例えば0.5秒とされる。入力トランスポートストリームの中から1つまたは複数のAVプログラムが記録のために選択される。選択されたトランスポートパケットは斜線を付して示されている。選択されたトランスポートパケットは、一般に図3(B)に示すように、不規則なタイミングで現れ、 Δt の間隔のタイムユニットTU i 毎のトランスポートパケットの数は変化する。なお、現実のトランスポートストリームにおいて、タイムユニットの時間長を約0.5秒と考える場合、タイムユニットに含まれるトランスポートパケットの数は、図3に示す例よりも、ずっと多いが、ここでは簡単化して図示している。

【0029】選択されたトランスポートパケットは、図4に示すように、間隔を詰めて記録媒体に記録される。この時、各トランスポートパケットには、それぞれのストリーム上の時刻を示すタイムスタンプ(Arrival Time Stamp)が付加される。Arrival Time Stampは、例えば、DVフォーマットで規定されているところのトランスポートパケットに付加される4バイト長のTSP_extra_headerと同様のものとされる。本明細書では、Arrival_Time_Sstampを含む4バイト長のヘッダの付加されたトランスポートパケットをソースパケットと呼ぶ。トランスポートパケットは、188バイト長であるので、ソースパケットは、192バイト長となる。

【0030】ソースパケットのシンタックスは、図5に示すように、TP_extra_header()とtransport_packet()で構成される。TP_extra_header()は、図6に示すように構成され、copy_permission_indicatorと、arrival_time_stampで構成される。

【0031】図4において、横軸は記録されたトランスポートストリームのバイト位置を示すアドレスである。また、横軸上にタイムユニット毎に最初に入力されたトランスポートパケットの先頭アドレスを示す。この例では、タイムユニットTU0, TU1, TU2では、それぞれ4個、3個、または6個のトランスポートパケットが記録されている。2つのタイムユニットにまたがって入力されるトランスポートパケットは、前側のタイムユニットに含められる。タイムユニットTU0, TU1, TU2の最初に入力されたトランスポートパケットの先頭アドレス、または、それらトランスポートパケットを含むソースパケットの先頭アドレスを、それぞれA(TU0), A(TU1), A(TU2)と表わすものとする。

【0032】図7は、タイムユニットマップ、すなわち記録されたトランスポートストリームのタイムユニット毎のデータの先頭アドレスのテーブルの例を示す。ここで、time_unit_addressは、記録されたストリーム上のタイムユニットの先頭データのアドレスを示す。タイムユニットマップでは、タイムユニット毎のデータ長 Δt a_time_unit_addressがテーブル化される。

【0033】この例においては、タイムユニットTU0のデータ長は、タイムユニットTU1の先頭のアドレスA(TU1)と、タイムユニットTU0の先頭のアドレスA(TU0)の差(A(TU1)-A(TU0))で表わされる。同様に、タイムユニットTU1のデータ長は、タイムユニットTU2の先頭のアドレスA(TU2)と、タイムユニットTU1の先頭のアドレスA(TU1)の差(A(TU2)-A(TU1))で表わされ、タイムユニットTU2のデータ長は、タイムユニットTU2の最後のアドレスend_addressと、タイムユニットTU2の先頭のアドレスA(TU2)の差(end_address-A(TU2))で表わされる。

【0034】次に、上述のエントリーポイントマップについて説明する。図8に示すトランスポートストリームは、図4に示したトランスポートストリームと同様のトランスポートストリームである。ここで斜線で示すトランスポートパケットにおいて、エントリーポイントが開始しているものとする。具体的には、エントリーポイントにおいて、MPEGビデオのシーケンスヘッダとIピクチャデータが開始しているものとする。所定のタイムユニットの中にエントリーポイントが存在する場合、そのタイムユニットのデータの先頭アドレスからエントリーポイントのアドレスまでのオフセットアドレスが計算される。すなわち、図8の例では、タイムユニットTU0とTU2にエントリーポイント(Iピクチャ)が存在する。そこで、タイムユニットTU0においては、その先頭のアドレスA(TU0)から、Iピクチャの先頭のアドレスI_start_addressまでの間隔aが、オフセットアドレスとして計算される。同時に、タイムユニットTU2においては、その先頭のアドレスA(TU2)から、Iピクチャの先頭のアドレスI_start_addressまでの間隔bがオフセットアドレスとして計算される。

【0035】図9は、エントリーポイントマップ、すなわちタイムユニット毎のエントリーポイントまでのオフセットアドレスのテーブルの例を示す。entry_point_flagは、対応するタイムユニットTU i にエントリーポイントが存在する時、「1」とされ、存在しない時、「0」とされる。entry_point_flagが「1」であるタイムユニットについて、そのタイムユニットのデータの先頭アドレスtime_unit_addressから、エントリーポイントのアドレスI_start_addressまでのオフセットアドレスI_start_offset_from_time_unit_addressは、次式に示すように計算される。

【0036】

$I_start_offset_from_time_unit_address$
 $= I_start_address - time_unit_address$

【0037】また、エントリーポイント毎に、エントリーポイントのIピクチャデータの終了アドレス $I_end_address$ 、エントリーポイントのIピクチャの次のPまたはIピクチャの終了アドレス $P1_end_address$ 、エントリーポイントのIピクチャの次の次のPまたはIピクチャの終了アドレス $P2_end_address$ が、次式に示すように計算される。

【0038】

$I_end_offset_address = I_end_address - I_start_address$

$P1_end_offset_address = P1_end_address - I_start_address$

$P2_end_offset_address = P2_end_address - I_start_address$

【0039】これらのアドレスの具体例を図10に示す。図10は、所定のタイムユニットの先頭からはじまるMPEGビデオデータを示す。ここで、I、P、BはそれぞれIピクチャ、Pピクチャ、またはBピクチャを表し、また添え字の数字は、ピクチャの表示順序を示す。このタイムユニットには、I2で示すエントリーポイントのIピクチャが存在する。また、IピクチャI2の次のPピクチャはP5であり、IピクチャI2の次の次のPピクチャは、P8である。この時、上記式で演算された $I_start_offset_from_time_unit_address$ 、 $I_end_offset_address$ 、 $P1_end_offset_address$ 、 $P2_end_offset_address$ は、図に示す関係になる。

【0040】すなわち、 $I_end_offset_address$ は、IピクチャI2の終了アドレス $I_end_address$ から、IピクチャI2の開始アドレス $I_start_address$ を減算した値とされている。 $P1_end_offset_address$ は、PピクチャP5の終了アドレス $P1_end_address$ から、IピクチャI2の開始アドレス $I_start_address$ を減算した値とされている。さらに $P2_end_offset_address$ は、PピクチャP8の終了アドレス $P2_end_address$ から、IピクチャI2の開始アドレス $I_start_address$ を減算した値とされている。

【0041】 $I_start_offset_from_time_unit_address$ は、エントリーポイントのアドレス $I_start_address$ から、タイムユニットのデータの先頭のアドレス $time_unit_address$ を減算した値とされている。

【0042】なお、記録するトランスポートストリームの中に複数のプログラムが含まれる場合、エントリーポイントの情報は、プログラム毎に区別して作成される。また、すべてのプログラムについて、エントリーポイントデータを用意できない場合を考慮して、エントリーポイントマップは、プログラム毎にエントリーポイントデータが存在するかどうかを示す情報(`parsed_program_flag`)を有する。

【0043】記録媒体に記録したトランスポートストリームを編集した場合、そのタイムユニットマップは変更(更新)される。次に、その方法を説明する。図11(A)は、図4に示すトランスポートストリームの先頭の2パケットと終わりの3パケットを消去する場合の例を示す。図11(B)は、このようにしてパケットが部分消去された後のトランスポートストリームを示す。図12は、図11(B)のトランスポートストリームのタイムユニットマップを示す。このようにタイムユニットの途中までのデータが消去された場合、最初のタイムユニットTU0の時間長(`first_time_unit_size`)が変化するので、これを書き換えられる。図11(B)の場合は、タイムユニットTU0の時間長が、タイムユニットTU1の先頭のパケットPbのタイムスタンプと、消去後のタイムユニットTU0の先頭のパケットPaのタイムスタンプの差分値に変更される。また、図12に示すように、タイムユニットTU0の`delta_time_unit_address`が、タイムユニットTU1の先頭のパケットPbのアドレスA(TU1)と、消去後のタイムユニットTU0の先頭のパケットPaのアドレスCの差分値(A(TU1)-C)に更新される。タイムユニットマップが変更された場合は、それに関するエントリーポイントマップも変更される。

【0044】次に上述のタイムユニットマップのシンタックスの例を図13と図14に示す。図13と図14は、それぞれタイムユニットマップのヘッダ部(`TimeUnitMapHeader()`)とデータ部(`TimeUnitMapData()`)を表す。タイムユニットマップをファイルとして記録する時は、ヘッダ部とデータ部を1つのファイルにして記録しても良いし、別々のファイルとして記録しても良い。`TimeUnitMapHeader()`の`start_time`、`end_time`は、それぞれ、このタイムユニットマップの開始時刻と終了時刻を示し、例えば、あるトランスポートストリームを記録する時の記録開始時刻と記録終了時刻を示す。`first_time_unit_size`は、最初のタイムユニットの時間長を示す。`time_unit_size`は、第2番目以降のタイムユニットの時間長を示す。`number_of_time_unit_entries`は、トランスポートストリームの中のタイムユニットの数を示す。`TimeUnitMapData()`には、`number_of_time_unit_entries`で示される数の`delta_time_unit_address`(図7)が書かれる。

【0045】また、上述のエントリーポイントマップのシンタックスの第1の例を、図15乃至図17に示す。図15は、エントリーポイントマップのヘッダ部(`EntryPointMapHeader()`)を表し、図16はエントリーポイントマップのデータ部(`EntryPointMapData()`)を表す。図17は、さらに図16の`entry_point_data()`のシンタックスを表わしている。エントリーポイントマップをファイルとして記録する時は、ヘッダ部とデータ部を1つのファイルにして記録しても良いし、別々のファイルとして記録しても良い。

【0046】図15のEntryPointMapHeader()のnumber_of_programsは、トランスポートストリームの中のプログラム数を示す。このシンタックスの第3行目から第6行目には、記録する各プログラムについて、エントリーマップテーブルが存在するかどうかを示す情報がある。第4行目のprogram_numberは、プログラムを特定(識別)する情報であり、対応するプログラムのPMT(Program Map Table)に書かれている情報である。第5行目のparsed_program_flagは、そのプログラムのエントリーポイントデータが存在するかどうかを示す。

【0047】第8行目から第10行目には、記録する各プログラムのPMTの情報が続く。MPEG2_TS_program_map_section()は、記録するトランスポートストリームの中から抽出した、MPEG2 systems規格で規定されているPMTである。ここで、NUMBER_OF_ParsedProgramsは、parsed_program_flagが「1」であるプログラムの数である。第8行目のNUMBER_OF_ParsedProgramsのループの中でデータが現れる順番は、第3行目のnumber_of_programsのループでparsed_program_flagが「1」であるprogram_numberが現れる順番である。

【0048】図16のEntryPointMapData()には、記録する各プログラムについてのエントリーポイントのデータが記述される。1つのタイムユニットについてのエントリーポイントのパラメータは、entry_point_flagとentry_point_data()である。1つのタイムユニットについてのentry_point_data()の内容は、図17に示すように、entry_point_time_stamp, l_start_offset_from_time_unit_address, l_end_offset_address, P1_end_offset_address, P2_end_offset_addressである。ここで、entry_point_time_stampは、エントリーポイントのトランスポートパケットのストリーム上の時刻、または、エントリーポイントのIピクチャのPTS(Presentation Time Stamp)に基づいて計算される。PTSは、MPEG2システムズ規格のPESパケットのヘッダに付加されている情報である。

【0049】また、上述のエントリーポイントマップのシンタックスの第2の例を、図18に示す。EntryPointMapHeader()とentry_point_data()の構成は、上述の第1の例における図15または図17に示す場合と同様である。この図18と図16を比較して明らかなように、各プログラムについてのエントリーポイントのデータの並び方が、図16の第1の例とは異なる。

【0050】次に、以下に示す状態の第1の例と第2の例のそれぞれの場合のエントリーマップのデータの並びの例を示す。ここでは、図19に示すように、トランスポートストリームの中に3個のプログラム(program#1, program#2, program#3)が多重化されていて、タイムユニットTUi(i=0, 1, 2, 3)毎に、各プログラムのエントリーポイントがあるものとする。この場合、各パラメータは次のようになる。

【0051】

```
number_of_time_unit_entries = 4
number_of_programs = 3
program_number = 1 : parsed_program_flag = 1
program_number = 2 : parsed_program_flag = 1
program_number = 3 : parsed_program_flag = 1
NUMBER_OF_ParsedPrograms = 3
```

【0052】図20は、第1の例(図16の例)の場合のエントリーポイントマップを示す。この場合、プログラム毎にエントリーポイントデータのリストが別れた形になる。すなわち、program#1のEntryPointMapDataは、図20(A)に示すように、タイムユニットTU0乃至TU3のそれぞれに、entry_point_dataとして、entry_point_data#1-1乃至entry_point_data#1-4が存在するため、entry_point_flagはそれぞれ「1」とされる。

【0053】なお、entry_point_data#A-Bは、program_number=AのB番目のエントリーポイントについてのentry_point_data()を表わす。

【0054】program#2のEntryPointMapDataは、図20(B)に示すように、タイムユニットTU1, TU3には、entry_point_dataが存在しないため、そのentry_point_flagは「0」とされる。これに対して、タイムユニットTU0, TU2においては、それぞれentry_point_data#2-1, entry_point_data#2-2が存在するため、そのentry_point_flagは「1」とされる。

【0055】さらに、program#3のEntryPointMapDataのタイムユニットTU0, TU2には、entry_point_dataが存在しないため、entry_point_flagは「0」とされる。タイムユニットTU1, TU3には、entry_point_data#3-1, entry_point_data#3-2がそれぞれ存在するため、そのentry_point_flagは「1」とされている。

【0056】これらのentry_point_flagと、entry_point_dataが、EntryPointMapDataに記述される。

【0057】また、図21は、第2の例(図18の例)の場合のエントリーポイントマップを示す。

【0058】この場合、タイムユニット毎に各プログラムのエントリーポイントデータが時間順に並ぶ形になり、エントリーポイントデータのリストは1つの形となる。すなわち、タイムユニットTU0において、3つのプログラムprogram#1乃至#3が記述され、それぞれについて、entry_point_flagと対応するentry_point_dataが記述される。この例では、program#3には、entry_point_dataが存在しないため、そのentry_point_flagは「0」とされ、program#1, #2については、entry_point_data#1-1, #2-1が存在するため、そのentry_point_flagは「1」とされている。

【0059】その他のタイムユニットTU1乃至TU3においても、program#1乃至#3それぞれについて、entry_point_flagと、entry_point_dataが記述される。

【0060】次に、図22を用いて、タイムユニットマ

ップの別の例を説明する。図22(A)、(B)は、前述の図3(A)、(B)と同様の構成を示す。また、図22(C)は、図4と同様に、記録媒体上に記録されたソースパケットデータを示す。トランスポートパケットとソースパケットの関係は、図4に示すとおりである。図22(C)では、横軸は記録されたソースパケットの番号を示す。ソースパケット番号は、ソースパケットの記録された順番であり、ゼロから開始して、順番に1ずつインクリメントする。タイムユニット毎のデータの先頭アドレスA(TU0)、A(TU1)、A(TU2)は、ソースパケット番号単位でカウントして、それぞれ0、4、7となる。

【0061】図22(D)は、タイムユニットマップ、すなわち記録されたトランスポートストリームのタイムユニット毎のデータの先頭アドレスのテーブルの例を示す。ここで、RSPN(Relative Source Packet Number) time_unit_startは、記録されたストリーム上のタイムユニットの先頭データのアドレスを示す。タイムユニットマップでは、記録されたトランスポートストリームの先頭からのタイムユニットの順にRSPN_time_unit_startがテーブル化され、図22(D)の場合、A(TU0)、A(TU1)、A(TU2)の順に値が並ぶ。

【0062】次に上述のタイムユニットマップのシンタックスの例を図23と図24に示す。図23と図24は、それぞれタイムユニットマップのヘッダ部(TimeUnitMapHeader())とデータ部(TimeUnitMapData())を表す。タイムユニットマップをファイルとして記録する時は、ヘッダ部とデータ部を1つのファイルにして記録しても良いし、別々のファイルとして記録しても良い。

【0063】図23のTimeUnitMapHeader()のoffset_SPNは、記録されたトランスポートストリームの一番目のソースパケットの番号を示す。本実施の形態では、RSPN_time_unit_startは、offset_SPNに対する相対的なソースパケット番号で表す。タイムユニットマップをはじめて作成する時の、offset_SPNの初期値は、ゼロである。次のoffset_timeは、このタイムユニットマップの開始時刻を示すものであり、一番目のタイムユニットの開始時刻を示す。

【0064】time_unit_sizeは、このタイムユニットマップのタイムユニットの時間長を示す。

【0065】number_of_time_unit_entriesは、記録されたトランスポートストリームの中のタイムユニットの数を示す。

【0066】図24のTimeUnitMapData()には、number_of_time_unit_entriesで示される数のRSPN_time_unit_start(図22(D))が書かれる。

【0067】次に、記録されたトランスポートストリームを編集した場合の上述のタイムユニットマップの変更(更新)の方法を説明する。図25(A)は、図22(C)に示すトランスポートストリームの先頭の2パケット、すなわち先頭からCで示すアドレスのソースパケ

ットまでを消去する場合の例を示す。図25(B)は、このようにしてパケットが部分消去された後のトランスポートストリームを示す。

【0068】図26は、図25(A)と(B)の場合のトランスポートストリームのタイムユニットマップを示す。このようにタイムユニットの先頭部分のデータが消去された場合、消去された部分のデータを含むタイムユニットを参照するRSPN_time_unit_startのデータが消去される。図26(B)の場合は、TU0のRSPN_time_unit_startのデータが消去される。これに伴って、number_of_time_unit_entriesが、消去したRSPN_time_unit_startのデータの数だけデクリメントされる。

【0069】また、上述のoffset_SPNとoffset_timeが変更される。図26(A)の場合は、offset_SPNは、ゼロであり、また、offset_timeは、TU0のタイムユニットの開始時刻のゼロがセットされている。図26(B)の場合は、offset_SPNが、新しく先頭になったソースパケットの元々のソースパケットの番号のC(=2)に変更される。また、offset_timeが新しく先頭になったタイムユニット、すなわち、この例ではTU1の元々の開始時刻の ΔT に変更される。なお、編集前後で、time_unit_sizeは、変更されない。

【0070】タイムユニットマップが変更された場合は、それに関係するエントリーポイントマップも変更される。

【0071】次に、入力されたトランスポートストリームから、上述のテーブルを作成してトランスポートストリームとともに、記録媒体に記録する動画像記録装置1の構成例を図27に示す。

【0072】端子10から入力されるトランスポートストリームには、1つまたは複数のAVプログラムが多重化されている。端子22には、ユーザインタフェースによって選択されたAVプログラムのチャネル(サービス名)が入力される。ここで選択されるチャネル数は、1つでも複数でも良い。

【0073】PIDフィルタ11は、入力されたトランスポートストリームの中から、ストリーム解析部12により指定されたPID(Packet ID)のトランスポートパケットを取り出す。PIDは、トランスポートパケットのヘッダの固定位置にある13ビット長の符号であり、そのトランスポートパケットのペイロードにストアされているデータのタイプを表す。はじめにPIDフィルタ11は、PID=0x0000であるPAT(Program Association Table)のトランスポートパケットを取り出す。PATには、トランスポートストリームに多重化されている各プログラムのPMT(Program Map Table)のトランスポートパケットのPIDが書かれている。PIDフィルタ11から出力されるPATのトランスポートパケットは、ストリーム解析部12へ入力される。

【0074】カウンタ24は、記録するトランスポート

ストリームの先頭パケットから現在のパケットまでのパケット数を計数し、現在のパケットナンバーを、タイムユニットマップ作成部23とエントリーポイントマップ作成部16へ出力する。

【0075】ストリーム解析部12は、PCR(Program Clock Reference)を伝送するトランスポートパケットからPCRを抽出して、PLL部13へ出力する。PCRを伝送するトランスポートパケットのPIDが複数ある場合は、どれか1つのPIDのパケットからPCRが抽出される。PLL部13は、入力されたPCRに同期して、27MHzの周波数のクロックを生成し、そのクロックをタイムスタンプ発生部14に出力する。

【0076】タイムスタンプ発生部14は、入力されたクロックをカウントし、そのカウント値に対応したタイムスタンプ(Arrival_Time_Stamp)を生成する。このタイムスタンプは、最初に記録するトランスポートパケットのタイムスタンプをゼロとすれば、そのトランスポートストリームの記録後の経過時間を表すことになる。このタイムスタンプは、ストリーム解析部12、タイムスタンプ付加部15、およびタイムユニットマップ作成部23へ出力される。

【0077】タイムスタンプ付加部15は、PIDフィルタ11から入力されたトランスポートパケットに、その到着時刻を示すタイムスタンプを含むヘッダーを付加し、ソースパケット(図4)をファイルシステム部17へ出力する。

【0078】タイムユニットマップ作成部23は、カウンタ24から入力されるパケットナンバーと、タイムスタンプ発生部14から入力されるタイムスタンプに基づいて、上述のタイムユニットマップを作成する。作成されたタイムユニットマップは、エントリーポイントマップ作成部16とファイルシステム部17へ出力される。

【0079】ストリーム解析部12は、プログラム毎の次に示すプログラム情報をエントリーポイントマップ作成部16へ出力する。

- (1) プログラムのprogram_number
- (2) プログラムのPMTのトランスポートパケットのPID
- (3) プログラムを構成するビデオのトランスポートパケットのPIDとstream_type
- (4) プログラムを構成するオーディオのトランスポートパケットのPIDとstream_type
- (5) プログラムのPCRのPID

ここで、stream_typeは、PMTに書いてある内容であり、ビデオの場合、MPEG2/MPEG1などのストリームタイプを表し、またオーディオの場合、MPEG1/AC-3などのストリームタイプを表す。

【0080】ストリーム解析部12はまた、記録するストリームのエントリーポイントデータを作成し、エントリーポイントマップ作成部16へ入力する。エントリーポイントデータの内容は、図17に示すものである。な

お、エントリーポイントのタイムスタンプをエントリーポイントのPTSとする場合、PTSはストリーム解析部12が入力ストリームから取り出すので、タイムスタンプ発生部14により作成したタイムスタンプをストリーム解析部12へ入力する必要はない。

【0081】エントリーポイントマップ作成部16は、エントリーポイントデータをプログラム毎にテーブル化し、上述のエントリーポイントマップを作成し、ファイルシステム部17へ出力する。

【0082】次に、その動作について説明する。PIDフィルタ11は、端子10からトランスポートストリームが入力されると、PID=0x0000であるPIDを含むトランスポートパケットを抽出し、ストリーム解析部12に出力する。ストリーム解析部12は、この時、図28のフローチャートに示す処理を実行する。

【0083】ステップS11で、ストリーム解析部12は、PIDフィルタ11からPID=0x0000のトランスポートパケットを受信すると、そのPATから、端子22を介して指令された各プログラムのPMTのトランスポートパケットのPIDを取得する。

【0084】ステップS12で、ストリーム解析部12は、各プログラムのPMTのPIDをPIDフィルタ11にセットする。PIDフィルタ11は、これらPMTのPIDをもつトランスポートパケットを取り出すと、それをストリーム解析部12へ出力する。

【0085】ステップS13で、ストリーム解析部12は、PIDフィルタ11からPMTのトランスポートパケットを受信する。PMTには、そのプログラムを構成するビデオストリームやオーディオストリームをベイロードに持つトランスポートパケットのPIDやPCR(Program Clock Reference)を伝送しているパケットのPIDが書かれている。ストリーム解析部12は、ユーザインタフェースによって選択された各プログラムを構成するビデオストリームやオーディオストリームをベイロードに持つトランスポートパケットのPIDとPCRを伝送しているパケットのPIDをここで取得する。

【0086】ステップS14で、ストリーム解析部12は、ユーザインタフェースによって選択された各プログラムを構成するビデオストリームやオーディオストリームをベイロードに持つトランスポートパケットのPIDとPCRを伝送しているパケットのPIDを、PIDフィルタ11にセットする。

【0087】なお、あらかじめEPG(Electrical Program Guide)等を伝送するサービスインフォメーションのパケットのPIDがわかっている場合は、これらのPIDもまた、PIDフィルタ11にセットされ、それらPIDのパケットも、PIDフィルタ11から出力される。

【0088】このようにして、PIDフィルタ11により抽出されたトランスポートパケットは、カウンタ24、ストリーム解析部12およびタイムスタンプ付加部15

に供給される。カウンタ24は、記録するトランスポートストリームの先頭の packets から現在の packets までの packets 数を計数し、現在の packets ナンバを検知する。検知された現在の packets NO. は、タイムユニットマップ作成部23と、エン트리ポイントマップ作成部16へ供給される。

【0089】また、ストリーム解析部12は、入力されるトランスポート packets からPCRを抽出し、PLL部13へ供給する。PLL部13は、入力されたPCRに同期して、27MHzの周波数のクロックを生成し、タイムスタンプ発生部14に供給する。

【0090】タイムスタンプ発生部14は、入力されたクロックをカウントし、そのカウント値に対応するタイムスタンプを生成する。タイムスタンプ付加部15は、PIDフィルタ11から入力されたトランスポート packets に、その到着時刻を示す、タイムスタンプ発生部14が発生したタイムスタンプを付加し、ソース packets をファイルシステム部17に供給する。

【0091】タイムユニットマップ作成部23は、カウンタ24から入力される packets ナンバと、タイムスタンプ発生部14から入力されるタイムスタンプに基づいて、図7に示したようなタイムユニット毎のtime_unit_addressと、delta_time_unit_addressとを対応させたタイムユニットマップを作成し、エン트리ポイントマップ作成部16と、ファイルシステム部17へ供給する。または、タイムユニットマップ作成部23は、図22(D)に示したようなタイムユニット毎のRSPN_time_unit_startを対応づけたタイムユニットマップを作成し、エン트리ポイントマップ作成部16と、ファイルシステム部17へ供給する。

【0092】ストリーム解析部12はまた、プログラム毎の上述したプログラム情報を、エン트리ポイントマップ作成部16へ供給する。

【0093】このため、ストリーム解析部12は、図29と図30に示すような、エン트리ポイントの解析処理を実行する。

【0094】ステップS31でストリーム解析部12は、記録するプログラムのビデオのPIDと、そのstream_typeをPIDフィルタ11にセットする。これにより、PIDフィルタ11から、指定したビデオの packets が、ストリーム解析部12に供給される。

【0095】ステップS32でストリーム解析部12は、ビデオ packets のポインタvppを初期化し、vpp=0とする。ポインタvppは、現在処理している上記PIDのビデオ packets の順番を表す。

【0096】ステップS33でストリーム解析部12は、ビデオ packets のポインタvppをインクリメントする(例えば、1だけ増加する)。

【0097】ステップS34で、ストリーム解析部12は、ペイロードの中のストリームに、MPEGビデオのsequence_header_code(32ビット長で"0x000001B3"の符号)が含まれているか否かを調べる。sequence_header_codeが含まれていない時は、処理はステップS33へ戻る。

【0098】ステップS34で、ペイロードにsequence_header_codeが含まれていると判定された時は、ステップS35へ進み、ストリーム解析部12は、sequence_header_codeを含む packets (最初のIピクチャの packets) のアドレスをI_start_addressとする(図10)。

【0099】ステップS36でストリーム解析部12は、ビデオ packets のポインタvppをインクリメントする。

【0100】ステップS37で、ストリーム解析部12は、上記Iピクチャのデータが終了したか否かを調べる。Iピクチャのデータがまだ終了していない場合、処理はステップS36へ戻る。Iピクチャのデータが終了した場合、処理はステップS38へ進む。

【0101】ステップS38で、ストリーム解析部12は、Iピクチャが終了する packets のアドレスをI_end_addressとする(図10)。以上により、最初のIピクチャのアドレスが決定されたことになる。

【0102】ストリーム解析部12は、ステップS39で(ビデオポインタvppはインクリメントしないで)、次のビデオ packets がシーケンスヘッダコードを含んでいるか否かを調べる。 packets がシーケンスヘッダコードを含んでいる場合、処理はステップS47へ進む。 packets がシーケンスヘッダコードを含んでいない場合、処理はステップS40へ進む。

【0103】ストリーム解析部12は、ステップS40でビデオ packets のポインタvppをインクリメントする。

【0104】ストリーム解析部12は、ステップS41で、PピクチャまたはIピクチャが終了したかどうかを調べる。PピクチャまたはIピクチャが終了していない場合、処理はステップS39へ戻る。PピクチャまたはIピクチャが終了している場合、処理はステップS42へ進む。

【0105】ストリーム解析部12は、ステップS42で、PまたはIピクチャが終了する packets のアドレスをP1_end_addressとする(図10)。以上により、Iピクチャの次の最初のPピクチャまたはIピクチャのアドレスが決定されたことになる。

【0106】ストリーム解析部12は、ステップS43で(ビデオポインタvppはインクリメントしないで)、次のビデオ packets がシーケンスヘッダコードを含んでいるか否かを調べる。ビデオ packets がシーケンスヘッダコードを含んでいる場合、処理はステップS47へ進む。ビデオ packets がシーケンスヘッダコードを含んでいない場合、処理はステップS44へ進む。

【0107】ストリーム解析部12は、ステップS44

でビデオパケットのポインタvppをインクリメントする。

【0108】ストリーム解析部12は、ステップS45でPピクチャまたはIピクチャが終了したかどうかを調べる。PピクチャまたはIピクチャが終了していない場合、処理はステップS43へ戻る。PピクチャまたはIピクチャが終了している場合、処理はステップS46へ進む。

【0109】ストリーム解析部12は、ステップS46で、PまたはIピクチャが終了するパケットのアドレスを、P2_end_addressとする(図10)。以上により、Iピクチャの次の次のPピクチャまたはIピクチャのアドレスが決定されたことになる。

【0110】ストリーム解析部12は、ステップS47でl_start_address, l_end_address, P1_end_address, P2_end_addressのアドレスを、エン트리ポイントマップ作成部16へ出力する。なお、この時、P1_end_addressとP2_end_addressの少くとも一方は存在しない場合もある。

【0111】ストリーム解析部12は、ステップS48で、現在のパケットが最後の入力パケットであるかどうかを判定する。現在のパケットが最後のパケットでない場合、処理はステップS33へ戻る。現在のパケットが最後のパケットである場合、処理は終了される。

【0112】以上のビデオストリームの解析は、記録するトランスポートストリームの中に複数のプログラムがある場合は、それぞれのプログラムのビデオパケットに対して行なわれる。

【0113】ストリーム解析部12は、以上のようにしてエン트리ポイントデータを生成すると、これをエン트리ポイントマップ作成部16に供給する。エン트리ポイントマップ作成部16は、ストリーム解析部12より供給されたエン트리ポイントデータを、プログラム毎にテーブル化し、図9に示すようなエン트리ポイントマップを作成し、ファイルシステム部17に供給する。

【0114】以上のようにして、ファイルシステム部17には、タイムスタンプ付加部15によりタイムスタンプが付加されたトランスポートストリームと、その特徴点を表わす特徴点データとしてのタイムユニットマップと、エン트리ポイントマップが、タイムユニットマップ作成部23とエン트리ポイントマップ作成部16からそれぞれ供給される。ファイルシステム部17は、トランスポートストリームと、それに対応する特徴点データをファイル化する。

【0115】図31は、このファイル構造の例を表わしている。この例においては、トランスポートストリームファイルの中に、3個のプログラムが多重化されている。同図に示すように、エン트리ポイントマップは、タイムユニットマップに従属する構成とされている。そして、各エン트리ポイントマップは、プログラム毎に

それぞれ次のデータを有する。

- (1) プログラムのprogram_number
- (2) プログラムのPMTのトランスポートパケットのPID
- (3) プログラムを構成するビデオのトランスポートパケットのPIDとstream_type
- (4) プログラムを構成するオーディオのトランスポートパケットのPIDとstream_type
- (5) プログラムのPCRのPID
- (6) エントリーポイントのリスト

【0116】ファイルシステム部17により生成されたファイルは、誤り訂正部18に供給され、誤り訂正符号が付加された後、変調部19に供給され、所定的方式で変調される。変調部19より出力された信号は、書き込み部20に供給され、記録媒体21に書き込まれる。

【0117】以上のようにして、トランスポートストリームとその特徴点データが、記録媒体21に記録される。

【0118】以上においては、タイムユニットマップとエン트리ポイントマップを、トランスポートストリームから作成するようにしたが、例えば、動画像記録装置自身が、トランスポートストリームを多重化し、生成するような場合、その多重化動作時に、タイムユニットマップとエン트리ポイントマップを、作成するようにすることもできる。図32は、この場合の構成例を表わしている。

【0119】すなわち、図32の例においては、多重化部40に複数(n個)のプログラムの、ビデオとオーディオのエレメンタリーストリーム#1乃至#nが入力されている。システムタイムクロック部42は、27MHzの周波数のシステムタイムクロックをカウントし、タイムスタンプを生成し、コントローラ41とタイムユニットマップ作成部43に出力している。コントローラ41は、多重化部40に入力された各エレメンタリーストリームを解析し、多重化部40が、MPEG2システム規格のT-STD(Transport Stream System Target Decoder)を満たして、トランスポートストリームを多重化するように、多重化部40を制御する。

【0120】コントローラ41は、多重化部40から出力される、トランスポートパケットの数を示すパケットナンバーを、タイムユニットマップ作成部43とエン트리ポイントマップ作成部44に出力する。タイムユニットマップ作成部43は、コントローラ41より入力されるパケットナンバーと、システムタイムクロック42より入力されるタイムスタンプに基づいて、タイムユニットマップを生成する。

【0121】コントローラ41はまた、プログラム情報とエン트리ポイントデータとを、エン트리ポイントマップ作成部44に出力する。エン트리ポイントマップ作成部44は、コントローラ41より供給される、パケットナンバー、プログラム情報、およびエン트리ポイント

ントデータ、並びにタイムユニットマップ作成部43より供給されるタイムユニットマップに基づいて、エントリーポイントマップを生成する。

【0122】多重化部40より出力されたトランスポートストリーム、タイムユニットマップ作成部43により作成されたタイムユニットマップ、およびエントリーポイントマップ作成部44により作成されたエントリーポイントマップは、それぞれ、図27に示したファイルシステム部17に供給される。ファイルシステム部17乃至記録媒体21までの構成は、図27に示した場合と同様である。

【0123】この図32に示すような構成の動画像記録装置1においては、コントローラ41が、多重化部40により多重化されるエレメンタリーストリームから、プログラム情報と、エントリーポイントデータを生成し、エントリーポイントマップ作成部44に出力する。また、コントローラ41は、システムタイムクロック42より入力されるタイムスタンプに対応するパケットナンバを、タイムユニットマップ作成部43とエントリーポイントマップ作成部44に出力する。

【0124】タイムユニットマップ作成部43は、コントローラ41から入力されるパケットナンバと、システムタイムクロック42より入力されるタイムスタンプに基づいて、タイムユニットマップを作成する。同様に、エントリーポイントマップ作成部44は、コントローラ41より入力されるパケットナンバ、プログラム情報、およびエントリーポイントデータ、並びにタイムユニットマップ作成部43より入力されるタイムユニットマップに基づいて、エントリーポイントマップを作成する。

【0125】そして、作成されたトランスポートストリーム、タイムユニットマップおよびエントリーポイントマップは、図27に示した場合と同様に、ファイルシステム部17によりファイル化され、誤り訂正部18により誤り訂正分が付加される。そして変調部19によりさらに変調された後、書き込み部20により、記録媒体21に記録される。

【0126】次に、以上のようにして、トランスポートストリームファイルと、そのストリームの特徴点データが記録された記録媒体21を再生する動画像再生装置について説明する。図33は、このような動画像再生装置51の構成例を表わしている。読み出し部61は、記録媒体21に記録されているデータを読み出し、復調部62に出力する。復調部62は、読み出し部61より入力されたデータを復調して、誤り訂正部63に出力する。誤り訂正部63は、復調部62より入力されたデータの誤りを訂正し、ファイルシステム部64に供給する。

【0127】ファイルシステム部64は、誤り訂正部63より入力されたデータを、トランスポートストリームファイルと、特徴点データとに分離し、ストリームファイルをバッファ65に供給するとともに、特徴点データ

を再生制御部71に出力する。再生制御部71は、端子73からユーザインタフェースを介して、ユーザより入力された指令に対応して、読み出し部61、デマルチプレキサ69、およびAVデコーダ70を制御する。

【0128】バッファ65は、ストリームファイルからarrival_time_stampに格納されている値を初期値として、カウンタ68に供給し、セットさせる。カウンタ68は、システムタイムクロック部67が発生する27MHzの周波数のクロックを、バッファ65からの初期値を基準としてカウントし、カウント値を比較部66に供給する。

【0129】比較部66は、カウンタ68より供給されたカウンタの値と、バッファ65より供給されたトランスポートパケットに含まれているarrival_time_stampの値を比較し、両者の値が一致したとき、そのトランスポートパケットをデマルチプレキサ69に出力する。

【0130】デマルチプレキサ69は、比較部66より入力されたトランスポートストリームファイルから、再生制御部71からの指令に対応するチャンネルのビデオデータと、オーディオデータとを抽出し、AVデコーダ70に出力する。AVデコーダ70は、デマルチプレキサ69より入力された、ビデオデータとオーディオデータをデコードし、端子72から出力する。

【0131】次に、その動作について説明する。記録媒体21には、図27（または図32）の動画像記録装置1で記録したトランスポートストリームファイルと、そのストリームの特徴点データが記録されている。トランスポートストリームファイルには、1つまたは複数のプログラムが多重化されている。

【0132】はじめに再生制御部71は、読み出し部61に対して、ストリームの特徴点データを読み出すように指示する。このとき、読み出し部61は、記録媒体21からストリームの特徴点データを読み出し、復調部62に出力する。復調部62は、入力されたデータを復調し、誤り訂正部63に出力する。誤り訂正部63は、入力されたデータの誤りを訂正し、ファイルシステム部64に供給する。ファイルシステム部64は、入力されたストリーム特徴点データを再生制御部71に出力する。

【0133】端子73からは、ユーザインタフェースによって再生を指定されたプログラム番号が入力され、それが再生制御部71へ入力される。再生制御部71は、そのプログラムのPMTのトランスポートパケットのPID、プログラムを構成するビデオのトランスポートパケットのPIDとstream_type、プログラムを構成するオーディオのトランスポートパケットのPIDとstream_type、並びにPCRのPIDを、特徴点データから読み出し、デマルチプレキサ69とAVデコーダ70へ出力する。

【0134】さらに、再生制御部71は、読み出し部61に対して、トランスポートストリームファイルを読み出すように指示する。この指令に対応して、読み出し部

61は、記録媒体21からトランスポートストリームファイルを読み出す。このデータは、上述した場合と同様に復調部62、誤り訂正部63、ファイルシステム部64の処理を経て、バッファ65へ入力される。

【0135】バッファ65は、入力されたトランスポートストリームファイルから、arrival_time_stampに記憶されている値を読み出し、カウンタ68に初期値として供給し、セットさせる。カウンタ68は、この初期値を基準として、システムタイムクロック部67が発生するクロックをカウントし、そのカウント値を比較部66に供給する。比較部66は、バッファ65より供給されるトランスポートストリームファイルから、arrival_time_stampの値を読み出し、カウンタ68から供給されるカウンタ値と比較する。両者の値が一致するタイミングで、比較部66は、トランスポートストリームファイルを、デマルチプレクサ69に出力する。

【0136】デマルチプレクサ69は、ユーザインタフェースにより指定されたプログラムを構成するビデオとオーディオのトランスポートパケットを、入力されたトランスポートストリームから分離し、それをAVデコーダ70へ入力する。AVデコーダ70は、ビデオストリームとオーディオストリームを復号し、再生ビデオ信号と再生オーディオ信号として端子72から出力する。

【0137】ユーザインタフェースによってランダムアクセス再生が指示された場合、再生制御部71は、内部

に記憶されているストリームの特徴点データの内容に基づいて、記録媒体21からのデータの読み出し位置を決定し、ランダムアクセス制御情報を読み出し部61へ入力する。例えば、ユーザによって選択されたプログラムを所定の時刻から途中再生する場合、再生制御部71は、タイムユニットマップに基づいて、指定された時刻に対応するトランスポートストリームのアドレスを計算し、そのアドレスからデータを読み出すように読み出し部61へ指示する。以下に、その手順を説明する。

【0138】はじめに、図7に示したタイムユニットマップの場合を説明する。ゼロ番目のタイムユニットTU0の先頭データの時刻をstart_timeとすれば、N番目(N>0)のタイムユニットの先頭データの時刻は、(start_time+first_time_unit_size+(N-1)*time_unit_size)となる。ユーザから指定された時刻よりもタイムユニットの先頭データの時刻が大きくなるタイムユニットの番号がわかったら、その番号のタイムユニットからデータを読み出せば良いことがわかる。

【0139】この場合、記録されたストリーム上の0番目のタイムユニットの先頭データのアドレスを0とすれば、N番目のタイムユニットの先頭データのアドレスtime_unit_address(N)は、次の様に計算できる。

【0140】

【数1】

$$\text{time_unit_address}(N) = \sum_{i=0}^{N-1} \text{delta_time_unit_address}(i)$$

【0141】次に、図22(D)に示したタイムユニットマップの場合を説明する。

【0142】この場合、N番目(N>=0)のタイムユニットの先頭データの時刻は、(offset_time+N* time_unit_size)となる。ユーザから指定された時刻よりもタイムユニットの先頭データの時刻が大きくなるタイムユニットの番号がわかったら、その番号のタイムユニットからデータを読み出せば良いことがわかる。N番目のタイムユニットの先頭データのソースパケット番号は、(RSPN_time_unit_start(N)-offset_SPN)となる。ここで、RSPN_time_unit_start(N)は、N番目のタイムユニットについてのRSPN_time_unit_startの値である。

【0143】また、ユーザによって選択されたプログラムに対応するエン트리ポイントマップのデータが存在する場合、再生制御部71は、エン트리ポイントデータに基づいて、特殊再生を制御できる。例えば、高速再生の場合、再生制御部71は、エン트리ポイント毎のアドレスのストリームデータを順次連続して読み出すように読み出し部61へ指示する。

【0144】図34は、この場合の再生制御部71の動作を表わしている。再生制御部71は、ステップS61で、ユーザからの指令に対応して、内蔵するメモリに、

再生するプログラムのprogram_numberをセットする。

【0145】再生制御部71は、ステップS62で、paesed_program_flagから、そのプログラムのエン트리ポイントデータが存在するか否かを調べる。存在する(paesed_program_flag=1である)場合は、ステップS63へ進む。エン트리ポイントデータが存在しない場合は、エン트리ポイントマップを使用したデータアクセスはできないので、処理は終了される。

【0146】再生制御部71は、ステップS63で、ユーザにより指定された時刻から読み出し開始するタイムユニットの番号TNを上述のようにして計算する。すなわち、図7に示したタイムユニットマップの場合、start_time+first_time_unit_size+(N-1)*time_unit_sizeの値(タイムユニットの先頭の時刻)が、指定された時刻よりも大きくなるタイムユニットの番号TNが計算される。または、図22(D)に示したタイムユニットマップの場合、(offset_time+N* time_unit_size)が、指定された時刻よりも大きくなるタイムユニットの番号TNが計算される。

【0147】再生制御部71は、ステップS64で、TN番目のタイムユニットに、そのプログラムのエン트리ポイントが存在するか否かを、entry_point_flagから調

べる。エントリーポイントが存在する(entry_point_flag=1である)場合は、ステップS65へ進み、存在しない場合は、ステップS67へ進む。

【0148】再生制御部71は、エントリーポイントが存在する場合、ステップS65で、entry_point_data()からエントリーポイントのストリームデータを読み出すアドレスを計算する。ストリームデータの読み出し開始アドレスは、l_start_addressであり、読み出し終了アドレスは、l_end_address、P1_end_address、またはP2_end_addressである。

【0149】再生制御部71は、ステップS66で、ステップS65で計算したアドレスに基づいて、エントリーポイントのストリームデータを読み出すように読み出し部61に指示する。読み出し部61はこの指示に対応して読み出し動作を実行する。

【0150】再生制御部71は、ステップS67で、番号TNをインクリメントする。再生制御部67は、ステップS68で、処理の終了が指令されたか否かを判定し、処理の終了が指令されていない場合は、ステップS64へ戻り、そうでない場合は処理を終了する。

【0151】読み出し部61は、指定されたランダムアクセスポイントからデータを読み出す。読み出されたデータは、復調部62、誤り訂正部63、ファイルシステム部64バッファ65、比較部66の処理を経て、デマルチプレクサ69へ入力され、AVデコーダ70で復号され、出力される。

【0152】このステップS63の計算処理の詳細について、図7に示したタイムユニットマップの場合を、図35と図36のフローチャートを参照してさらに説明する。ステップS81において、再生制御部71に、端子73からprogram_numberと、再生開始時刻Tstが入力されると、ステップS82において、再生制御部71は、ステップS81で入力された再生開始時刻Tstが、特徴点データに含まれる、トランスポートストリームの開始時刻start_time(図3(B))と等しいか否かを判定する。再生開始時刻Tstが開始時刻start_timeと等しい場合には、ステップS86に進み、再生制御部71は、タイムユニットの番号を表わす変数Nに0を設定し、そのタイムユニット(0番目のタイムユニット)のtime_unit_address(N)に0を設定する。

【0153】これに対して、ステップS82において、再生開始時刻Tstが開始時刻start_timeと等しくないと判定された場合、ステップS83に進み、再生制御部71は、タイムユニットマップのヘッダ部を読み込み、ステップS84において、次の不等式を満たす最小の値Nを計算する。

【0154】 $Tst \leq start_time + first_time_unit_size + (N-1) \times time_unit_size$

ステップS85において、再生制御部71は、タイムユニットマップのデータに基づいて、数1で示す式に従っ

て、time_unit_address(N)を演算する。

【0155】N番目のタイムユニットの先頭データの時刻time_unit_address(N)が求められた時、ステップS87において、再生制御部71は、N番目のタイムユニットのアドレスtime_unit_address(N)からのデータ読み出しを、読み出し部61に指示する。

【0156】読み出し部61は、再生制御部71からの指令に対応して、ステップS88において、アドレスtime_unit_address(N)からのトランスポートストリームを記録媒体21から読み出す。読み出されたデータは、復調部62、誤り訂正部63、ファイルシステム部64、バッファ65、比較部66を介して、デマルチプレクサ69に供給される。

【0157】ステップS89において、再生制御部71は、デマルチプレクサ69に対して、ユーザより再生が指示された、プログラムのprogram_numberを出力する。デマルチプレクサ69は、ステップS90において、再生制御部71より指示された、program_numberのプログラムのトランスポートパケットを分離し、AVデコーダ70に出力する。ステップS91において、AVデコーダ70は、デマルチプレクサ69より入力されたデータをデコードし、端子72から出力する。

【0158】また、図34のフローチャートのステップS63の計算処理の詳細について、図22(D)に示したタイムユニットマップの場合を、図37のフローチャートを参照してさらに説明する。

【0159】図37のフローチャートは、図35と図36のフローチャートのステップS82、S84、S85を、それぞれステップS102、S104、S105へ変更したものである。これら以外は、図37のフローチャートは、図35と図36のフローチャートと同じである。以下では、ステップS102、S104、S105の各ステップについてのみ説明する。

【0160】ステップS102では、再生開始時刻Tstがタイムユニットマップの開始時刻offset_timeと比較される。

【0161】ステップS104においては、次の不等式を満たす最小の値Nが計算される。 $Tst \leq offset_time + N \times time_unit_size$

【0162】ステップS85においては、再生制御部71が、タイムユニットマップのデータに基づいて、次に示す式に従って、time_unit_address(N)を演算する。 $time_unit_address(N) = RSPN_time_unit_start(N) - offset_SPN$

【0163】上述した一連の処理は、ハードウェアにより実行させることもできるが、ソフトウェアにより実行させることもできる。一連の処理をソフトウェアにより実行させる場合には、そのソフトウェアを構成するプログラムが、専用のハードウェアとしての動画像記録再生装置に組み込まれているコンピュータ、または、各種の

プログラムをインストールすることで、各種の機能を実行することが可能な、例えば汎用のパーソナルコンピュータなどにインストールされる。

【0164】次に、図38を参照して、上述した一連の処理を実行するプログラムをコンピュータにインストールし、コンピュータによって実行可能な状態とするために用いられる記録媒体について、そのコンピュータが汎用のパーソナルコンピュータである場合を例として説明する。

【0165】プログラムは、図38(A)に示すように、コンピュータ301に内蔵されている記録媒体としてのハードディスク302や半導体メモリ303に予めインストールした状態でユーザに提供することができる。

【0166】あるいはまた、プログラムは、図38(B)に示すように、フロッピー（登録商標）ディスク311、CD-ROM(Compact Disk-Read Only Disk)312、MO(Magneto-Optical)ディスク313、DVD(Digital Versatile Disk)314、磁気ディスク315、半導体メモリ316などの記録媒体に、一時的あるいは永続的に格納し、パッケージソフトウェアとして提供することができる。

【0167】さらに、プログラムは、図38(C)に示すように、ダウンロードサイト321から、デジタル衛星放送用の人工衛星322を介して、コンピュータ323に無線で転送したり、ローカルエリアネットワーク、インターネットといったネットワーク131を介して、コンピュータ323に有線で転送し、コンピュータ323において、内蔵するハードディスクなどに格納させることができる。

【0168】本明細書における記録媒体とは、これら全ての記録媒体を含む広義の概念を意味するものである。

【0169】また、本明細書において、記録媒体により提供されるプログラムを記述するステップは、記載された順序に沿って時系列的に行われる処理はもちろん、必ずしも時系列的に処理されなくとも、並列的あるいは個別に実行される処理をも含むものである。

【0170】なお、本明細書において、システムとは、複数の装置により構成される装置全体を表すものである。

【0171】このように、1つまたは複数のトランスポートストリームが記録された記録媒体にランダムアクセスして再生する場合、Iピクチャやオーディオフレームの開始位置を効率よくサーチすることができるので、ユーザ入力に対して応答の速いランダムアクセス再生をすることができる。

【0172】

【発明の効果】以上の如く、本発明の第1のデータ処理装置、請求項9に記載のデータ処理方法、および記録媒体のプログラムによれば、符号化ストリームを所定のタ

イムユニット毎に区分し、区分された符号化ストリームのタイムユニット毎のデータのアドレスを示すタイムユニットマップを作成するようにしたので、応答の速いランダムアクセスが可能となる。

【0173】本発明の第2のデータ再生装置、データ再生方法、および記録媒体プログラムによれば、記録媒体から再生されたタイムユニットマップに基づいて、記録媒体に記録されている符号化ストリームを、任意の位置から再生するようにしたので、迅速なランダム生成が可能となる。

【図面の簡単な説明】

【図1】従来のトランスポートストリームのパケットを説明する図である。

【図2】従来の記録されるトランスポートストリームを説明する図である。

【図3】本発明のトランスポートストリームを説明する図である。

【図4】本発明の記録されるトランスポートストリームを説明する図である。

【図5】Source Packetのシンタックスを示す図である。

【図6】TP_extra_headerのシンタックスを示す図である。

【図7】タイムユニットマップの例を示す図である。

【図8】タイムユニット毎のオフセットアドレスを説明する図である。

【図9】エン트리ポイントマップの例を示す図である。

【図10】エン트리ポイントデータを説明する図である。

【図11】データの消去を説明する図である。

【図12】データを消去した時のタイムユニットマップの例を示す図である。

【図13】TimeUnitMapHeader()のシンタックスを示す図である。

【図14】TimeUnitMapData()のシンタックスを示す図である。

【図15】EntryPointMapHeader()のシンタックスを示す図である。

【図16】EntryPointMapData()のシンタックスを示す図である。

【図17】entry point data()のシンタックスを示す図である。

【図18】EntryPointMapData()のシンタックスを示す図である。

【図19】トランスポートストリームファイルのエン트리ポイントを説明する図である。

【図20】EntryPointMapDataの例を示す図である。

【図21】EntryPointMapDataの例を示す図である。

【図22】本発明のトランスポートストリームを説明す

る図である。

【図23】TimeUnitMapHeader()のシンタクスを示す図である。

【図24】TimeUnitMapData()のシンタクスを示す図である。

【図25】データの消去を説明する図である。

【図26】データを消去したときのタイムユニットマップの例を示す図である。

【図27】本発明を適用した動画像記録装置の構成例を示すブロック図である。

【図28】図27の動画像記録装置の動作を説明するフローチャートである。

【図29】図27の動画像記録装置の動作を説明するフローチャートである。

【図30】図27の動画像記録装置の動作を説明するフローチャートである。

【図31】トランスポートストリームファイルのタイムユニットマップとエン트리ポイントマップの関係を説明する図である。

【図32】本発明を適用した動画像記録装置の他の構成例を示すブロック図である。

【図33】本発明を適用した動画像再生装置の構成例を

示すブロック図である。

【図34】図33の動画像再生装置の動作を説明するフローチャートである。

【図35】図33の動画像再生装置の動作を説明するフローチャートである。

【図36】図33の動画像再生装置の動作を説明するフローチャートである。

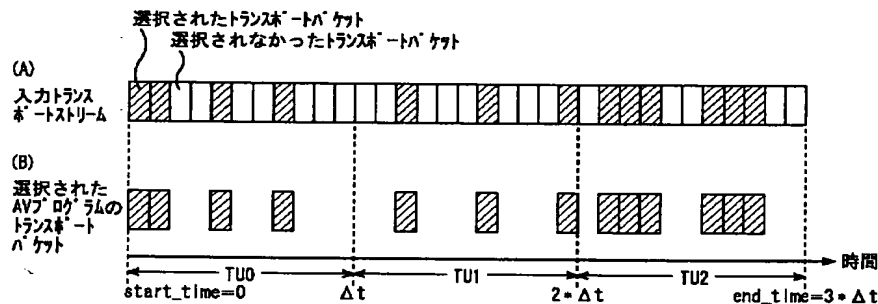
【図37】図33の動画像再生装置の動作を説明するフローチャートである。

【図38】記録媒体を説明する図である。

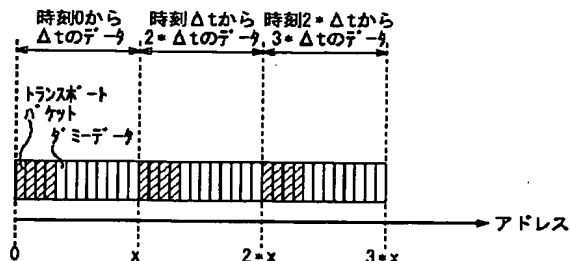
【符号の説明】

1 動画像記録装置, 11 PIDフィルタ, 12 ストリーム解析部, 14 タイムスタンプ発生部, 15 タイムスタンプ付加部, 16 エン트리ポイントマップ作成部, 17 ファイルシステム部, 21 記録媒体, 23 タイムユニットマップ作成部, 24 カウンタ, 40 多重化部, 41 コントローラ, 42 システムタイムクロック部, 43 タイムユニットマップ作成部, 44 エン트리ポイントマップ作成部, 61 読み出し部, 69 デマルチプレクサ, 70 AVデコーダ, 71 再生制御部

【図1】



【図2】

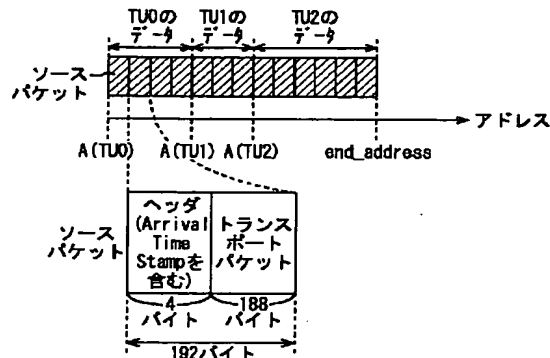


【図12】

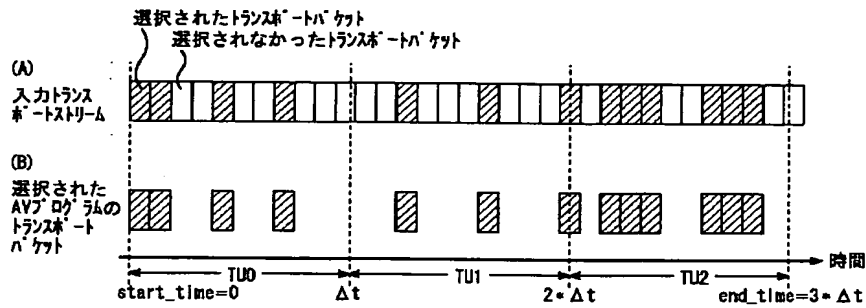
タイムユニット	delta_time_unit_address
TU0	A(TU1)-C
TU1	A(TU2)-A(TU1)
TU2	D-A(TU2)

部分消去後のタイムユニットマップ

【図4】



【図3】



【図5】

Syntax	No. of bits	Mnemonics
source_packet {		
TP_extra_header ()	4byte	
transport_packet ()	188byte	
}		

【図6】

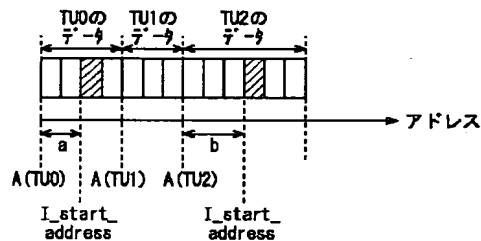
Syntax	No. of bits	Mnemonics
TP_extra_header {		
copy_permission_indicator	2	uimsbf
arrival_time_stamp	30	uimsbf
}		

【図7】

タイムユニット	time_unit_address	delta_time_unit_address
TU0	A(TU0)	A(TU1) - A(TU0)
TU1	A(TU1)	A(TU2) - A(TU1)
TU2	A(TU2)	end_address - A(TU2)

タイムユニットマップ

【図8】

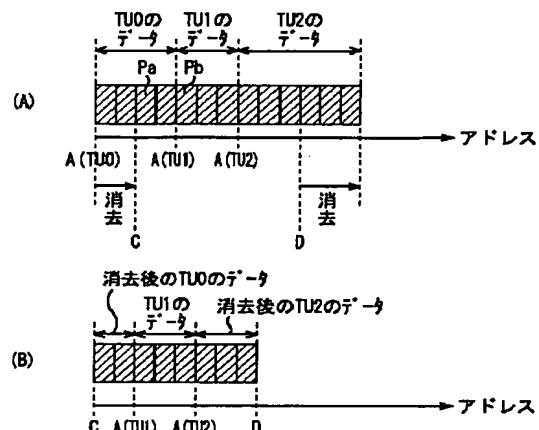


【図9】

タイムユニット	entry_point_flag	I_start_offset_from_time_unit_address
TU0	1	a
TU1	0	-
TU2	1	b

エントリポイントマップ

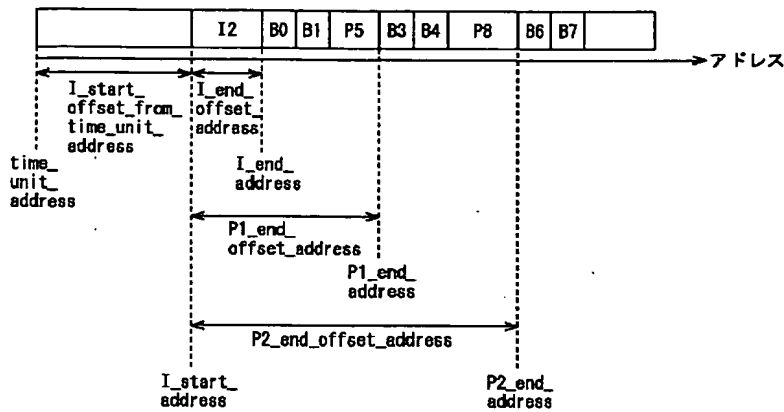
【図11】



【図13】

Syntax	Bits
TimeUnitMapHeader {	
1 start_time	N1
2 end_time	N2
3 first_time_unit_size	N3
4 time_unit_size	N4
5 number_of_time_unit_entries	N5
6 }	

【図10】



【図14】

TimeUnitMapData 0		
Syntax		Bits
1	TimeUnitMapData 0 {	
2	for (i=0; i<number_of_time_unit_entries; i++) {	
3	delta_time_unit_address	N6
4	}	
5	}	

【図16】

EntryPointMapData 0		
Syntax		Bits
1	EntryPointMapData 0 {	
2	for (i=0; i<NUMBER_OF_ParsedPrograms; i++) {	
3	for (j=0; j<number_of_time_unit_entries; j++) {	
4	entry_point_flag	1
5	entry_point_data 0	
6	}	
7	}	

【図18】

EntryPointMapData 0		
Syntax		Bits
1	EntryPointMapData 0 {	
2	for (i=0; i<number_of_time_unit_entries; i++) {	
3	for (j=0; j<NUMBER_OF_ParsedPrograms; j++) {	
4	entry_point_flag	1
5	entry_point_data 0	
6	}	
7	}	
8	}	

【図15】

EntryPointMapHeader 0		
Syntax		Bits
1	EntryPointMapHeader 0 {	
2	number_of_programs	K1
3	for (i=0; i<number_of_programs; i++) {	
4	program_number	16
5	parsed_program_flag	1
6	}	
7		
8	for (i=0; i<NUMBER_OF_ParsedPrograms; i++) {	
9	MPEG_TS_program_map_section 0	
10	}	
11	}	

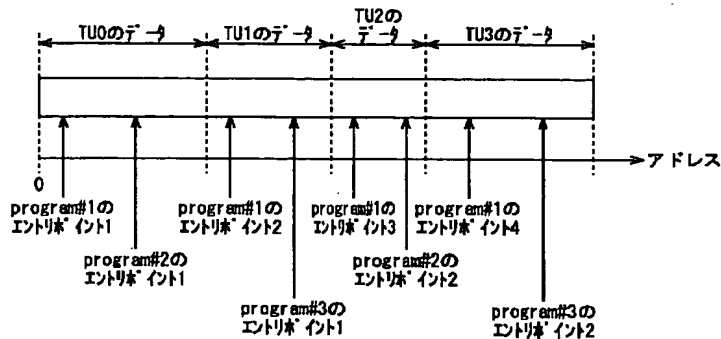
【図17】

entry_point_data 0		
Syntax		Bits
1	entry_point_data 0 {	
2	entry_point_time_stamp	K2
3	I_start_offset_from_time_unit_address	K3
4	I_end_offset_address	K4
5	P1_end_offset_address	K5
6	P2_end_offset_address	K6
7	}	

【図23】

Syntax	No. of bits
TimeUnitMapHeader 0 {	
offset_SPN	32
offset_time	32
time_unit_size	32
number_of_time_unit_entries	32
}	

【図19】



【図20】

(A)

タイムユニット	entry_point_flag	entry_point_data
TU0	1	entry_point_data#1-1
TU1	1	entry_point_data#1-2
TU2	1	entry_point_data#1-3
TU3	1	entry_point_data#1-4

(B)

タイムユニット	entry_point_flag	entry_point_data
TU0	1	entry_point_data#2-1
TU1	0	-(dummy data)
TU2	1	entry_point_data#2-2
TU3	0	-

(C)

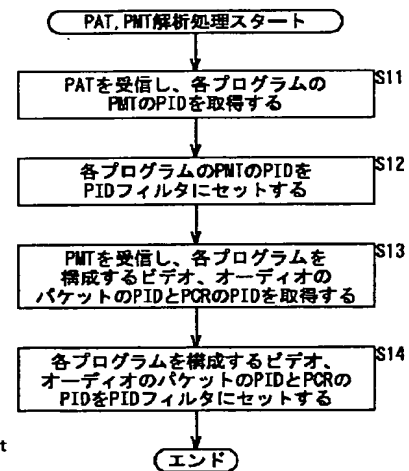
タイムユニット	entry_point_flag	entry_point_data
TU0	0	-
TU1	1	entry_point_data#3-1
TU2	0	-
TU3	1	entry_point_data#3-2

【図21】

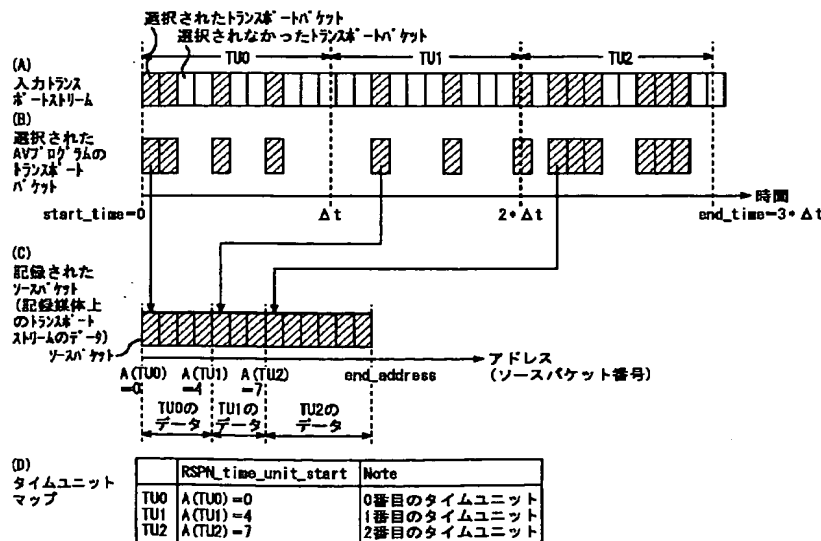
program#1, program#2, program#3のEntryPointMapData

タイムユニット	Program number	entry_point_flag	entry_point_data
TU0	#1	1	entry_point_data#1-1
	#2	1	entry_point_data#2-1
	#3	0	-
TU1	#1	1	entry_point_data#1-2
	#2	0	-
	#3	1	entry_point_data#3-1
TU2	#1	1	entry_point_data#1-3
	#2	1	entry_point_data#2-2
	#3	0	-
TU3	#1	1	entry_point_data#1-4
	#2	0	-
	#3	1	entry_point_data#3-2

【図28】



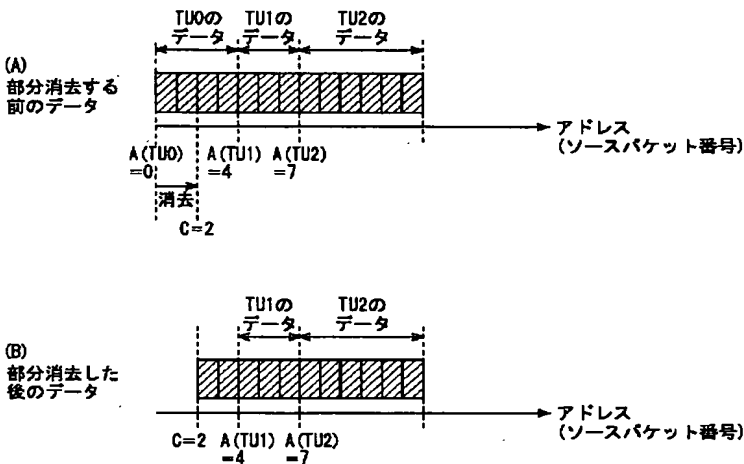
【図22】



【図24】

Syntax	No. of bits
TimeUnitMapData0 {	
for (k=0; k<number_of_time_unit_entries; k++) {	
RSPN_time_unit_start	32
}	
}	

【図25】



【図26】

(A) 部分消去前のタイムユニットマップ

	RSPN_time_unit_start	Note
TU0	A(TU0)=0	0番目のタイムユニット
TU1	A(TU1)=4	1番目のタイムユニット
TU2	A(TU2)=7	2番目のタイムユニット

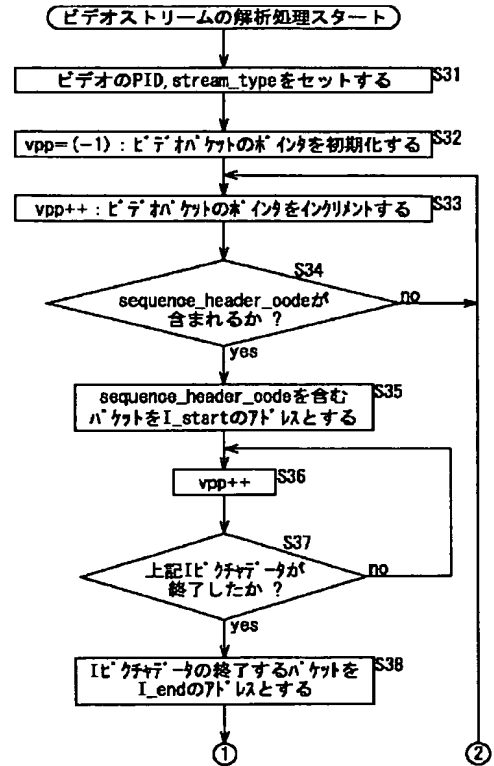
-offset_SPN=0
-offset_time=0: (TU0の開始時刻)
-time_unit_size=ΔT
-number_of_time_unit_entries=3

(B) 部分消去後のタイムユニットマップ

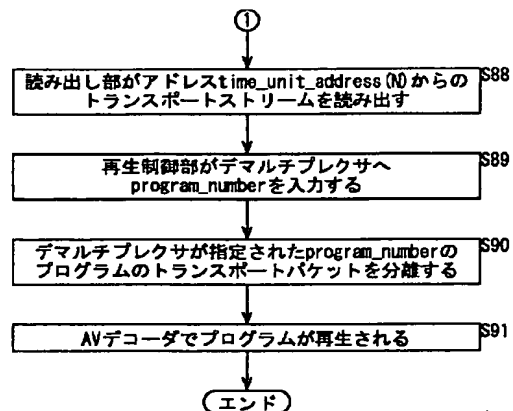
	RSPN_time_unit_start	Note
TU1	A(TU1)=4	新0番目のタイムユニット
TU2	A(TU2)=7	新1番目のタイムユニット

-offset_SPN=C(-2)
-offset_time=ΔT: (TU1の開始時刻)
-time_unit_size=ΔT
-number_of_time_unit_entries=2

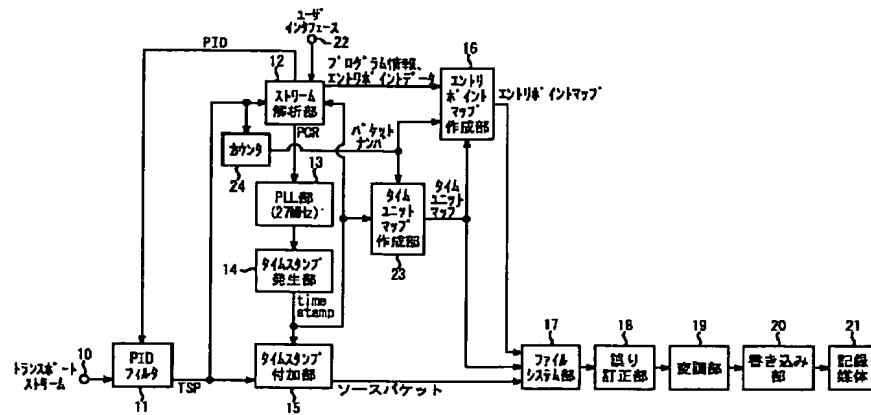
【図29】



【図36】

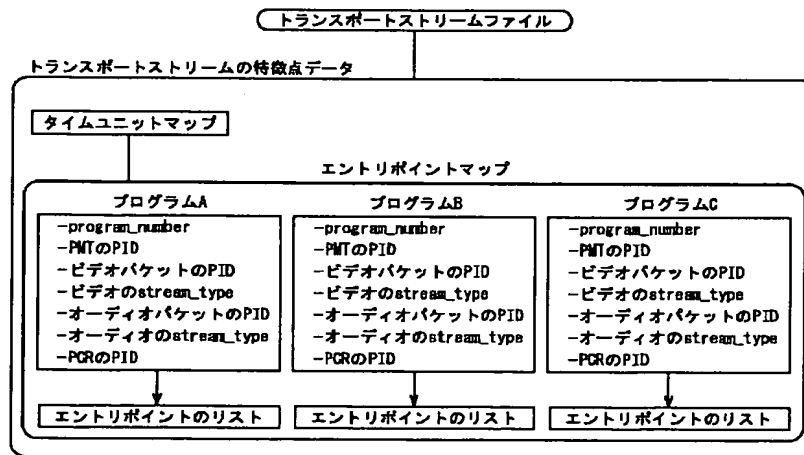


【図27】

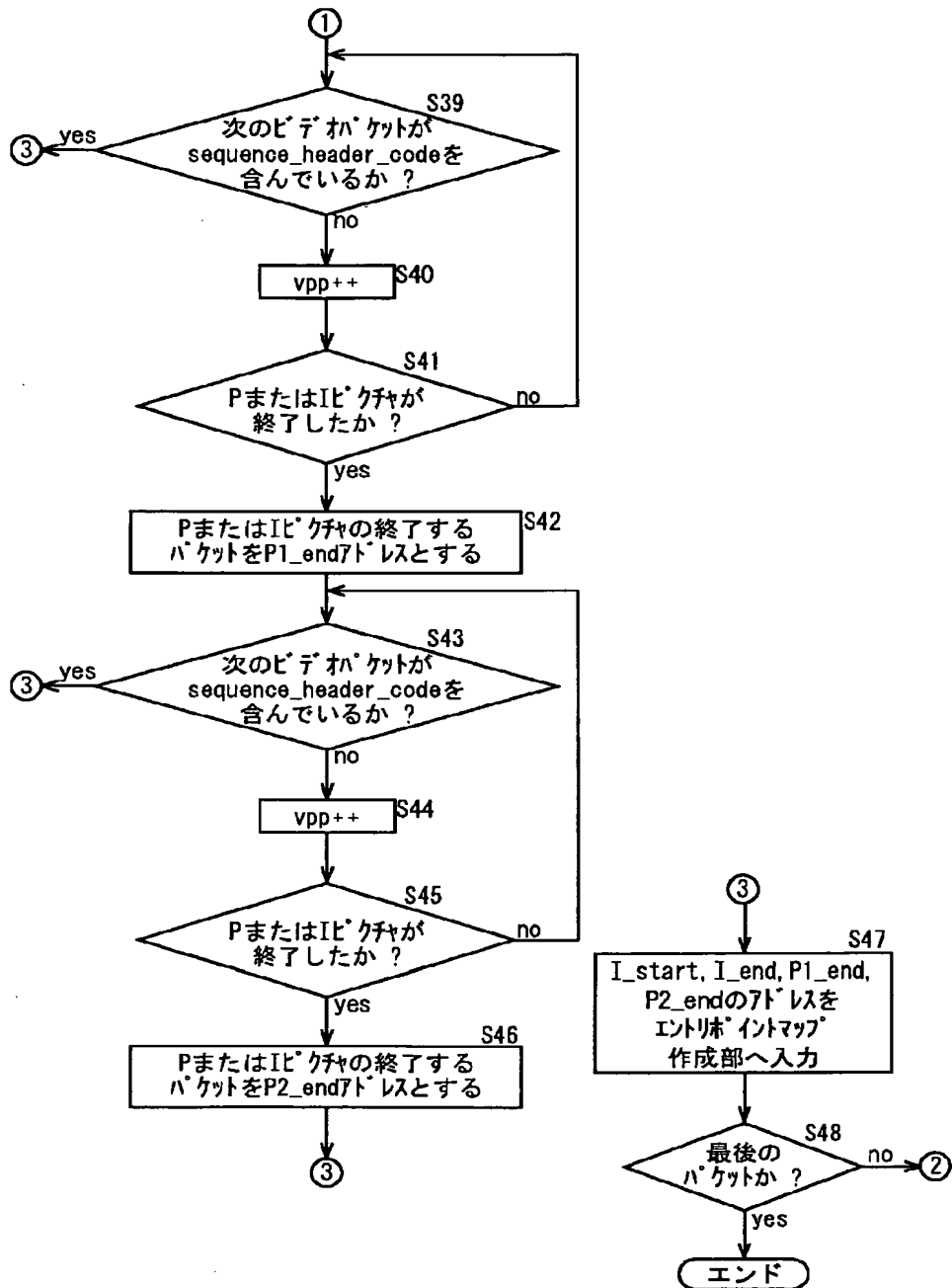


動画記録装置 1

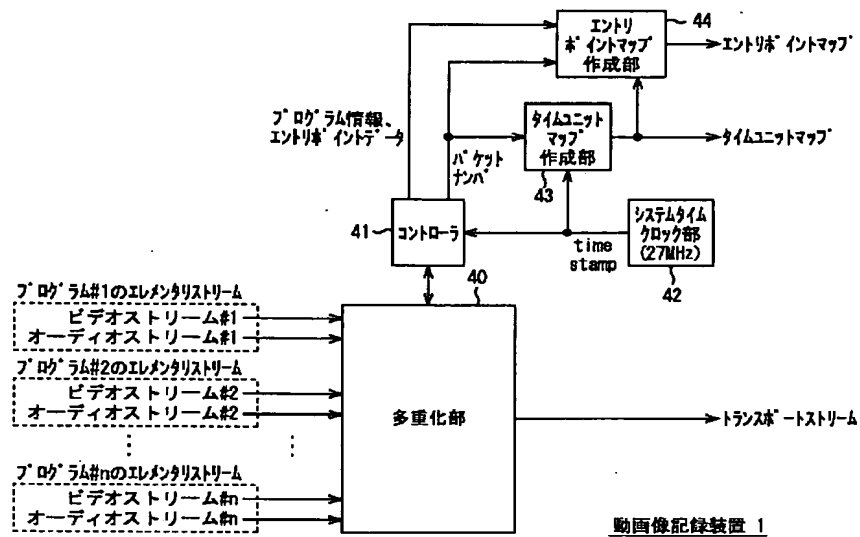
【図31】



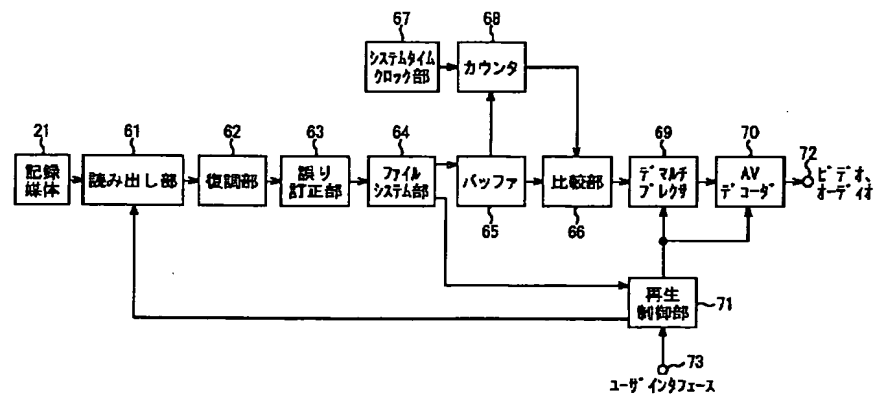
【図30】



【図32】

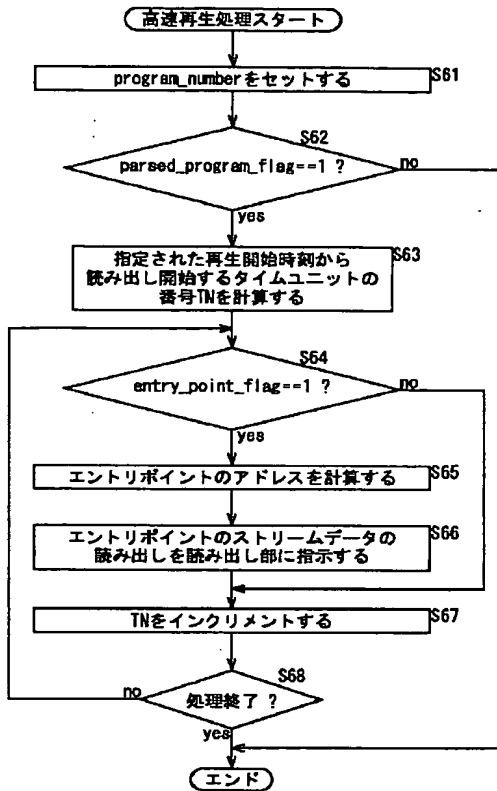


【図33】

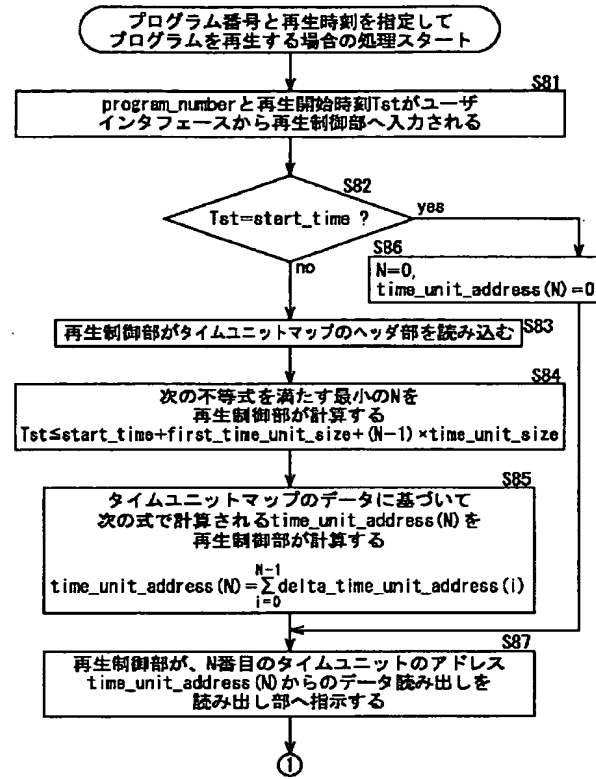


動画再生装置 51

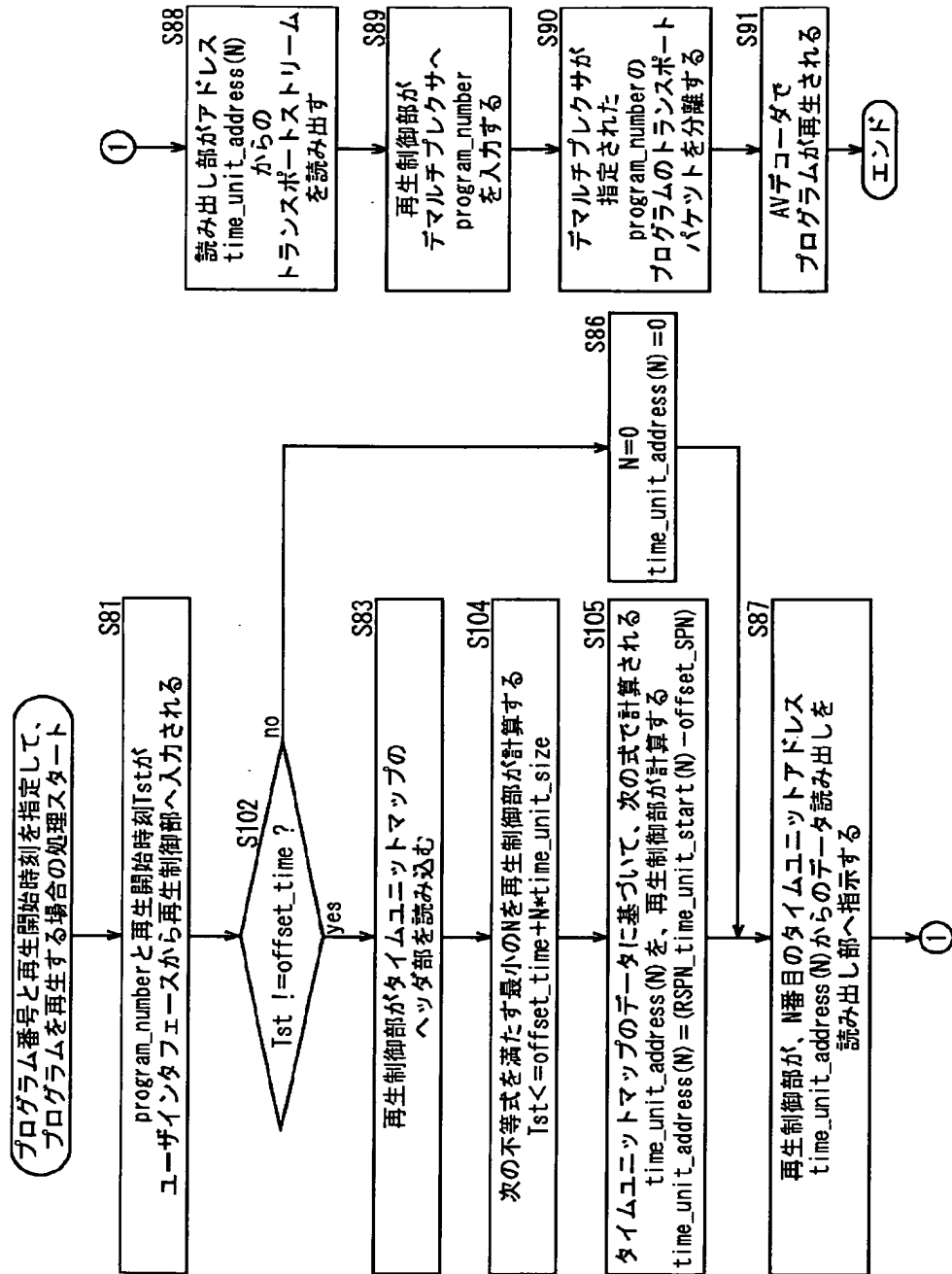
【図34】



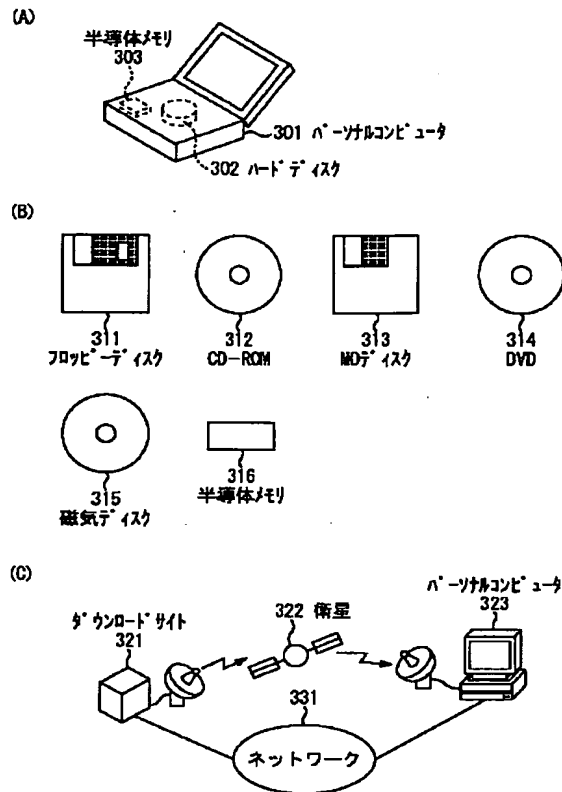
【図35】



【図37】



【図38】



【手続補正書】

【提出日】平成12年4月17日(2000.4.17)

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】全文

【補正方法】変更

【補正内容】

【書類名】明細書

【発明の名称】データ処理装置および方法、データ再生装置および方法、並びに記録媒体

【特許請求の範囲】

【請求項1】 入力された符号化ストリームに含まれるデータを処理するデータ処理装置において、入力された前記符号化ストリームを、所定のタイムユニット毎に区分する区分手段と、前記区分手段により区分された前記符号化ストリームのタイムユニット毎のデータのアドレスを示すタイムユニットマップを作成する第1の作成手段とを備えることを特徴とするデータ処理装置。

【請求項2】 前記タイムユニットマップは、前記タイ

ムユニット毎のデータのアドレスとして、タイムユニット毎の先頭のパケットのパケット番号を保持することを特徴とする請求項1に記載のデータ処理装置。

【請求項3】 前記タイムユニットマップは、前記タイムユニットの先頭のデータのアドレスと、そのタイムユニット内に含まれるデータ量に対応するアドレスの間隔とを対応して保持することを特徴とする請求項1に記載のデータ処理装置。

【請求項4】 前記符号化ストリームとともに、前記タイムユニットマップをファイル化するファイル化手段をさらに備えることを特徴とする請求項1に記載のデータ処理装置。

【請求項5】 前記ファイル化手段によりファイル化されたデータを記録媒体に記録する記録手段をさらに備えることを特徴とする請求項4に記載のデータ処理装置。

【請求項6】 前記第1の作成手段は、前記符号化ストリームが編集されたとき、前記タイムユニットマップを変更することを特徴とする請求項1に記載のデータ処理装置。

【請求項7】 前記符号化ストリームのエントリポイン

トの位置を示す、前記タイムユニットマップに従属するエントリポイントマップを作成する第2の作成手段をさらに備えることを特徴とする請求項1に記載のデータ処理装置。

【請求項8】 前記エントリポイントマップは、前記エントリポイントの前記タイムユニットの先頭からのアドレスの間隔を含むことを特徴とする請求項7に記載のデータ処理装置。

【請求項9】 前記エントリポイントは、前記タイムユニット内に含まれるIピクチャの先頭のアドレスを表し、

前記エントリポイントマップは、前記Iピクチャの先頭のアドレスから、前記Iピクチャの終了アドレスまでの間隔、または前記Iピクチャが含まれるタイムユニット内の、それより後のPピクチャの終了アドレスまでの間隔を含むことを特徴とする請求項8に記載のデータ処理装置。

【請求項10】 前記エントリポイントマップは、前記タイムユニットにおける前記エントリポイントの有無を表すフラグを含むことを特徴とする請求項7に記載のデータ処理装置。

【請求項11】 前記符号化ストリームが、複数のプログラムを多重化して生成された符号化ストリームである場合、前記第2の作成手段は、前記エントリポイントマップを、前記プログラム毎に作成することを特徴とする請求項7に記載のデータ処理装置。

【請求項12】 前記第2の作成手段は、前記符号化ストリームが編集されたとき、前記エントリポイントマップを変更することを特徴とする請求項7に記載のデータ処理装置。

【請求項13】 入力された符号化ストリームに含まれるデータを処理するデータ処理装置のデータ処理方法において、

入力された前記符号化ストリームを、所定のタイムユニット毎に区分する区分ステップと、
前記区分ステップの処理により区分された前記符号化ストリームのタイムユニット毎のデータのアドレスを示すタイムユニットマップを作成する作成ステップとを含むことを特徴とするデータ処理方法。

【請求項14】 入力された符号化ストリームに含まれるデータを処理するプログラムであって、 入力された前記符号化ストリームを、所定のタイムユニット毎に区分する区分ステップと、
前記区分ステップの処理により区分された前記符号化ストリームのタイムユニット毎のデータのアドレスを示すタイムユニットマップを作成する作成ステップとを含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項15】 記録媒体に記録されている、符号化ストリームのタイムユニット毎のデータのアドレスを示す

タイムユニットマップを再生する第1の再生手段と、
前記第1の再生手段により再生された、前記タイムユニットマップに基づいて、前記記録媒体に記録されている前記符号化ストリームを、任意の位置から再生する第2の再生手段とを備えることを特徴とするデータ再生装置。

【請求項16】 前記第1の再生手段は、前記符号化ストリームのエントリポイントの位置を示す、前記タイムユニットマップに従属するエントリポイントマップもさらに再生し、

前記第2の再生手段は、前記第1の再生手段により再生された、前記タイムユニットマップとエントリポイントマップに基づいて、前記記録媒体に記録されている前記符号化ストリームを、任意の位置から再生することを特徴とする請求項15に記載のデータ再生装置。

【請求項17】 記録媒体に記録されている、符号化ストリームのタイムユニット毎のデータのアドレスを示すタイムユニットマップを再生する第1の再生ステップと、

前記第1の再生ステップの処理により再生された、前記タイムユニットマップに基づいて、前記記録媒体に記録されている前記符号化ストリームを、任意の位置から再生する第2の再生ステップとを含むことを特徴とするデータ再生方法。

【請求項18】 記録媒体に記録されている、符号化ストリームのタイムユニット毎のデータのアドレスを示すタイムユニットマップを再生する第1の再生ステップと、

前記第1の再生ステップの処理により再生された、前記タイムユニットマップに基づいて、前記記録媒体に記録されている前記符号化ストリームを、任意の位置から再生する第2の再生ステップとを含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項19】 符号化ストリームとともに、前記符号化ストリームのタイムユニット毎のデータのアドレスを示すタイムユニットマップがファイル化されて記録されていることを特徴とする記録媒体。

【請求項20】 前記タイムユニットマップは、前記タイムユニット毎のデータのアドレスとして、タイムユニット毎の先頭のパケットのパケット番号を保持することを特徴とする請求項19に記載の記録媒体。

【請求項21】 前記タイムユニットマップは、前記タイムユニットの先頭のデータのアドレスと、そのタイムユニット内に含まれるデータ量に対応するアドレスの間隔とを対応して保持することを特徴とする請求項19に記載の記録媒体。

【請求項22】 前記符号化ストリームのエントリポイントの位置を示す、前記タイムユニットマップに従属するエントリポイントマップがさらに記録されていること

を特徴とする請求項19に記載の記録媒体。

【請求項23】 前記エントリポイントマップは、前記エントリポイントの前記タイムユニットの先頭からのアドレスの間隔を含むことを特徴とする請求項22に記載の記録媒体。

【請求項24】 前記エントリポイントは、Iピクチャの先頭のアドレスを表し、

前記エントリポイントマップは、前記Iピクチャの先頭のアドレスから、前記Iピクチャの終了アドレスまでの間隔、または前記Iピクチャと同一のタイムユニット内の、それより後のPピクチャの終了アドレスまでの間隔を含むことを特徴とする請求項23に記載の記録媒体。

【請求項25】 前記エントリポイントマップは、前記タイムユニットにおける前記エントリポイントの有無を表すフラグを含むことを特徴とする請求項22に記載の記録媒体。

【請求項26】 前記エントリポイントマップは、プログラム毎に作成されていることを特徴とする請求項22に記載の記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、データ処理装置および方法、並びに記録媒体に関し、特に符号化ストリームから特徴点情報を抽出することにより、複数のプログラムが多重化されている場合においても、迅速にランダムアクセスができるようにしたデータ処理装置および方法、並びに記録媒体に関する。

【0002】

【従来の技術】ヨーロッパのDVB(Digital Video Broadcast)や日本のデジタルBS放送などの多チャネルデジタルテレビジョン放送では、MPEG(Moving Picture Experts Group)2トランスポートストリームが使われる。トランスポートストリームは、トランスポートパケットが連続したストリームであり、トランスポートパケットは、例えば、MPEG2ビデオストリームやMPEG1オーディオストリームがパケット化されたものである。放送の電波で伝送される1本のトランスポートストリームには、1つまたは複数のAV(Audio Visual)プログラムが多重化されている。一般に、各チャネルのAVプログラムは、お互いに独立している。

【0003】したがって、放送で送られるトランスポートストリームを家庭の受信機でそのまま受信し、記録すれば、そのトランスポートストリームのすべてのチャネルのプログラムを同時に記録することができる。また、放送で送られるトランスポートストリームの中からユーザによって選択された幾つかのチャネルのAVプログラムのトランスポートストリームを分離したものを記録すれば、選択された任意の数のチャネルのプログラムを同時に記録することができる。

【0004】図1に従来のトランスポートストリームの

記録方法の例を示す。図1(A)は、複数のAVプログラムが多重化されたトランスポートストリームを示す。ここで横軸は、時間であり、 Δt の間隔のタイムユニットTUI($i = 0, 1, 2, \dots$)毎に区切られている。入力トランスポートストリームの中から1つまたは複数のAVプログラムが選択される。選択されたトランスポートパケットは斜線を施して示されている。選択されたトランスポートパケットは、一般に、図1(B)に示すように、不規則なタイミングで現れ、タイムユニットTUI毎のトランスポートパケットの数は変化する。

【0005】 Δt の間隔のタイムユニットTUI毎の選択されたトランスポートパケットは、図2に示すように、間隔を詰めて記録媒体に記録される。この時、各トランスポートパケットは、それぞれのストリーム上の時刻を示すタイムスタンプを付加して記録される。このタイムスタンプは、例えば、DV(Digital Video)フォーマット(Specification Of Consumer-Use Digital VCRs using 6.3mm magnetic tape(HD digital VCR conference)), PART7 DVB Specifications of Consumer-Use Digital VCR)で規定されているところのトランスポートパケットに付加される4バイト長のTSP_extra_headerと同様のものである。

【0006】図2において、横軸は記録されたトランスポートストリームのバイト位置を示すアドレスである。図1(B)に示すような可変ビットレートのトランスポートストリームが入力されると、記録装置は、図2に示すように、ダミーデータを入れて、固定の記録レートでデータを記録する。したがって、記録されたトランスポートストリームの時間の経過に対するデータ量は比例する。すなわち、タイムユニットあたりの記録データ量を x とすると、 n 番目($n = 0, 1, 2, \dots$)のタイムユニットの先頭データのバイト位置は、 n 倍の x となる。

【0007】

【発明が解決しようとする課題】このように、従来の記録方法は、ダミーデータを挿入して一定の記録レートにしているため、トランスポートストリームの記録効率が良くない。しかしながら、ダミーデータを挿入しないと、記録されたトランスポートストリームの時間の経過とファイルのデータ量が比例しなくなるので、トランスポートストリームの時間軸上の所定の位置のデータにアクセスする場合、データのアクセス性が悪くなる問題が発生する。

【0008】また、一般に、MPEG2ビデオのストリームでは、0.5秒くらいの間隔でIピクチャが符号化され、それ以外のピクチャはPピクチャまたはBピクチャとして符号化される。したがって、MPEG2ビデオのストリームが記録された記録媒体から、ビデオ信号を高速再生する場合は、Iピクチャをサーチしなければならない。ところが、デジタル放送等のトランスポートストリームが記録された記録媒体から、ランダムアクセスにより再生

を行う場合、Iピクチャの開始バイトを効率よくサーチすることが難しかった。すなわち、記録媒体上のトランスポートストリームのランダムなバイト位置から読み出したビデオストリームのシンタックスを解析し、Iピクチャやオーディオフィームの開始バイトがサーチされていた。そのため、場合によっては、Iピクチャのサーチに時間がかかってしまい、ユーザ入力に対して応答の速いランダムアクセス再生をすることが困難であった。

【0009】本発明は、このような状況に鑑みてなされたものであり、複数のプログラムが多重化されている場合においても、迅速に、ランダムアクセスができるようにするものである。

【0010】

【課題を解決するための手段】本発明のデータ処理装置は、入力された符号化ストリームを、所定のタイムユニット毎に区分する区分手段と、区分手段により区分された符号化ストリームのタイムユニット毎のデータのアドレスを示すタイムユニットマップを作成する第1の作成手段とを備えることを特徴とする。

【0011】前記タイムユニットマップは、タイムユニット毎のデータのアドレスとして、タイムユニット毎の先頭のパケットのパケット番号を保持するようにすることができる。

【0012】前記タイムユニットマップは、タイムユニットの先頭のデータのアドレスと、そのタイムユニット内に含まれるデータ量に対応するアドレスの間隔とを対応して保持するようにすることができる。

【0013】前記符号化ストリームとともに、タイムユニットマップをファイル化するファイル化手段をさらに備えるようにすることができる。

【0014】前記ファイル化手段によりファイル化されたデータを記録媒体に記録する記録手段をさらに備えるようにすることができる。

【0015】前記第1の作成手段は、符号化ストリームが編集されたとき、タイムユニットマップを変更するようにすることができる。

【0016】前記符号化ストリームのエントリポイントの位置を示す、タイムユニットマップに従属するエントリポイントマップを作成する第2の作成手段をさらに備えるようにすることができる。前記エントリポイントマップは、エントリポイントのタイムユニットの先頭からのアドレスの間隔を含むようにすることができる。前記エントリポイントは、タイムユニット内に含まれるIピクチャの先頭のアドレスを表し、エントリポイントマップは、Iピクチャの先頭のアドレスから、Iピクチャの終了アドレスまでの間隔、またはIピクチャが含まれるタイムユニット内の、それより後のPピクチャの終了アドレスまでの間隔を含むようにすることができる。前記エントリポイントマップは、タイムユニットにおけるエントリポイントの有無を表すフラグを含むようにするこ

とができる。

【0017】前記符号化ストリームが、複数のプログラムを多重化して生成された符号化ストリームである場合、第2の作成手段は、エントリポイントマップを、プログラム毎に作成するようにすることができる。

【0018】前記第2の作成手段は、符号化ストリームが編集されたとき、エントリポイントマップを変更するようにすることができる。

【0019】本発明のデータ処理方法は、入力された符号化ストリームを、所定のタイムユニット毎に区分する区分ステップと、区分ステップの処理により区分された符号化ストリームのタイムユニット毎のデータのアドレスを示すタイムユニットマップを作成する作成ステップとを含むことを特徴とする。

【0020】本発明の第1の記録媒体のプログラムは、入力された符号化ストリームを、所定のタイムユニット毎に区分する区分ステップと、区分ステップの処理により区分された符号化ストリームのタイムユニット毎のデータのアドレスを示すタイムユニットマップを作成する作成ステップとを含むことを特徴とする。

【0021】本発明のデータ再生装置は、記録媒体に記録されている、符号化ストリームのタイムユニット毎のデータのアドレスを示すタイムユニットマップを再生する第1の再生手段と、第1の再生手段により再生された、タイムユニットマップに基づいて、記録媒体に記録されている符号化ストリームを、任意の位置から再生する第2の再生手段とを備えることを特徴とする。

【0022】第1の再生手段は、符号化ストリームのエントリポイントの位置を示す、タイムユニットマップに従属するエントリポイントマップもさらに再生し、第2の再生手段は、第1の再生手段により再生された、タイムユニットマップとエントリポイントマップに基づいて、記録媒体に記録されている符号化ストリームを、任意の位置から再生するようにすることができる。

【0023】本発明のデータ再生方法は、記録媒体に記録されている、符号化ストリームのタイムユニット毎のデータのアドレスを示すタイムユニットマップを再生する第1の再生ステップと、第1の再生ステップの処理により再生された、タイムユニットマップに基づいて、記録媒体に記録されている符号化ストリームを、任意の位置から再生する第2の再生ステップとを含むことを特徴とする。

【0024】本発明の第2の記録媒体のプログラムは、記録媒体に記録されている、符号化ストリームのタイムユニット毎のデータのアドレスを示すタイムユニットマップを再生する第1の再生ステップと、第1の再生ステップの処理により再生された、タイムユニットマップに基づいて、記録媒体に記録されている符号化ストリームを、任意の位置から再生する第2の再生ステップとを含むことを特徴とする。本発明の記録媒体は、符号化スト

リームとともに、符号化ストリームのタイムユニット毎のデータのアドレスを示すタイムユニットマップがファイル化されて記録されていることを特徴とする。前記タイムユニットマップは、タイムユニット毎のデータのアドレスとして、タイムユニット毎の先頭のパケットのパケット番号を保持するようにすることができる。前記タイムユニットマップは、タイムユニットの先頭のデータのアドレスと、そのタイムユニット内に含まれるデータ量に対応するアドレスの間隔とを対応して保持するようにすることができる。前記符号化ストリームのエントリポイントの位置を示す、タイムユニットマップに従属するエントリポイントマップがさらに記録されているようにすることができる。前記エントリポイントマップは、エントリポイントのタイムユニットの先頭からのアドレスの間隔を含むようにすることができる。前記エントリポイントマップは、Iピクチャの先頭のアドレスを表し、エントリポイントマップは、Iピクチャの先頭のアドレスから、Iピクチャの終了アドレスまでの間隔、またはIピクチャと同一のタイムユニット内の、それより後のPピクチャの終了アドレスまでの間隔を含むようにすることができる。前記エントリポイントマップは、タイムユニットにおけるエントリポイントの有無を表すフラグを含むようにすることができる。エントリポイントマップは、プログラム毎に作成されているようにすることができる。本発明のデータ処理装置、データ処理方法、および、第1の記録媒体のプログラムにおいては、入力された符号化ストリームが、タイムユニット毎に区分され、そのタイムユニット毎のデータのアドレスを示すタイムユニットマップと、符号化ストリームのエントリポイントの位置を示すタイムユニットマップに従属するエントリポイントマップが作成される。

【0025】本発明のデータ処理装置、データ処理方法、および、第2の記録媒体のプログラムにおいては、記録媒体から符号化ストリームのタイムユニット毎のデータのアドレスを示すタイムユニットマップと、符号化ストリームのエントリポイントの位置を示すタイムユニットマップに従属するエントリポイントマップが再生され、それらに基づいて符号化ストリームが任意の位置から再生される。本発明の記録媒体においては、符号化ストリームとともに、符号化ストリームのタイムユニット毎のデータのアドレスを示すタイムユニットマップがファイル化されて記録されている。

【0026】

【発明の実施の形態】以下、符号化ストリームが1つまたは複数のプログラムが多重化されている多重化ストリームである場合を例として本発明を説明するが、本発明は、符号化ストリームがMPEGビデオストリーム等のエレメンタリーストリームであっても適用できるものである。

【0027】最初に、本発明の基本的な原理について説

明する。本発明の動画像記録装置は、1つまたは複数のプログラムが多重化されているトランスポートストリームをディスク、テープなどの記録媒体に記録する時に、ストリーム上の時間を所定のタイムユニット（単位時間）毎に区切り、タイムユニット毎のデータのストリーム上のアドレスを計算する。そして、このタイムユニット毎のデータのストリーム上のアドレスを示すタイムユニットマップが作成される。さらに、記録するトランスポートストリームのプログラム毎のエントリポイント（ランダムアクセスポイント）の場所を示すエントリポイントマップが作成される。エントリポイントマップは、タイムユニットマップに従属する構造を有する。このタイムユニットマップについて、以下に説明する。

【0028】図3は、複数のAVプログラムが多重化されたトランスポートストリームを示す。ここで横軸は、時間を示し、 Δt の間隔のタイムユニット TU_i ($i=0, 1, 2, \dots$) 毎に区切られている。文字 TU の後ろに続く数字 i は、タイムユニット TU の時間順序を示す。最初のオリジナル記録の時は、すべてのタイムユニット TU の時間長は同じ値 Δt である。値 Δt の大きさは、例えば0.5秒とされる。入力トランスポートストリームの中から1つまたは複数のAVプログラムが記録のために選択される。選択されたトランスポートパケットは斜線を付して示されている。選択されたトランスポートパケットは、一般に図3(B)に示すように、不規則なタイミングで現れ、 Δt の間隔のタイムユニット TU_i 毎のトランスポートパケットの数は変化する。なお、現実のトランスポートストリームにおいて、タイムユニットの時間長を約0.5秒と考える場合、タイムユニットに含まれるトランスポートパケットの数は、図3に示す例よりも、ずっと多いが、ここでは簡単化して図示している。

【0029】選択されたトランスポートパケットは、図4に示すように、間隔を詰めて記録媒体に記録される。この時、各トランスポートパケットには、それぞれのストリーム上の時刻を示すタイムスタンプ(Arrival Time Stamp)が付加される。Arrival Time Stampは、例えば、DVフォーマットで規定されているところのトランスポートパケットに付加される4バイト長のTSP_extra_headerと同様のものとされる。本明細書では、Arrival_Time_Sstampを含む4バイト長のヘッダの付加されたトランスポートパケットをソースパケットと呼ぶ。トランスポートパケットは、188バイト長であるので、ソースパケットは、192バイト長となる。

【0030】ソースパケットのシンタックスは、図5に示すように、TP_extra_header()とtransport_packet()で構成される。TP_extra_header()は、図6に示すように構成され、copy_permission_indicatorと、arrival_time_stampで構成される。

【0031】図4において、横軸は記録されたトランスポートストリームのバイト位置を示すアドレスである。

また、横軸上にタイムユニット毎に最初に入力されたトランスポートパケットの先頭アドレスを示す。この例では、タイムユニットTU0、TU1、TU2では、それぞれ4個、3個、または6個のトランスポートパケットが記録されている。2つのタイムユニットにまたがって入力されるトランスポートパケットは、前側のタイムユニットに含められる。タイムユニットTU0、TU1、TU2の最初に入力されたトランスポートパケットの先頭アドレス、または、それらトランスポートパケットを含むソースパケットの先頭アドレスを、それぞれA(TU0)、A(TU1)、A(TU2)と表わすものとする。

【0032】図7は、タイムユニットマップ、すなわち記録されたトランスポートストリームのタイムユニット毎のデータの先頭アドレスのテーブルの例を示す。ここで、time_unit_addressは、記録されたストリーム上のタイムユニットの先頭データのアドレスを示す。タイムユニットマップでは、タイムユニット毎のデータ長delta_time_unit_addressがテーブル化される。

【0033】この例においては、タイムユニットTU0のデータ長は、タイムユニットTU1の先頭のアドレスA(TU1)と、タイムユニットTU0の先頭のアドレスA(TU0)の差(A(TU1)-A(TU0))で表わされる。同様に、タイムユニットTU1のデータ長は、タイムユニットTU2の先頭のアドレスA(TU2)と、タイムユニットTU1の先頭のアドレスA(TU1)の差(A(TU2)-A(TU1))で表わされ、タイムユニットTU2のデータ長は、タイムユニットTU2の最後のアドレスend_addressと、タイムユニットTU2の先頭のアドレスA(TU2)の差(end_address-A(TU2))で表わされる。

【0034】次に、上述のエントリーポイントマップについて説明する。図8に示すトランスポートストリームは、図4に示したトランスポートストリームと同様のトランスポートストリームである。ここで斜線で示すトランスポートパケットにおいて、エントリーポイントが開始しているものとする。具体的には、エントリーポイントにおいて、MPEGビデオのシーケンスヘッダとIピクチャデータが開始しているものとする。所定のタイムユニットの中にエントリーポイントが存在する場合、そのタイムユニットのデータの先頭アドレスからエントリーポイントのアドレスまでのオフセットアドレスが計算される。すなわち、図8の例では、タイムユニットTU0とTU2にエントリーポイント(Iピクチャ)が存在する。そこで、タイムユニットTU0においては、その先頭のアドレスA(TU0)から、Iピクチャの先頭のアドレスI_start_addressまでの間隔aが、オフセットアドレスとして計算される。同時に、タイムユニットTU2においては、その先頭のアドレスA(TU2)から、Iピクチャの先頭のアドレスI_start_addressまでの間隔bがオフセットアドレスとして計算される。

【0035】図9は、エントリーポイントマップ、すな

わちタイムユニット毎のエントリーポイントまでのオフセットアドレスのテーブルの例を示す。entry_point_flagは、対応するタイムユニットTUiにエントリーポイントが存在する時、「1」とされ、存在しない時、「0」とされる。entry_point_flagが「1」であるタイムユニットについて、そのタイムユニットのデータの先頭アドレスtime_unit_addressから、エントリーポイントのアドレスI_start_addressまでのオフセットアドレスI_start_offset_from_time_unit_addressは、次式に示すように計算される。

【0036】

$$I_start_offset_from_time_unit_address = I_start_address - time_unit_address$$

【0037】また、エントリーポイント毎に、エントリーポイントのIピクチャデータの終了アドレスI_end_address、エントリーポイントのIピクチャの次のPまたはIピクチャの終了アドレスP1_end_address、エントリーポイントのIピクチャの次の次のPまたはIピクチャの終了アドレスP2_end_addressが、次式に示すように計算される。

【0038】

$$I_end_offset_address = I_end_address - I_start_address$$

$$P1_end_offset_address = P1_end_address - I_start_address$$

$$P2_end_offset_address = P2_end_address - I_start_address$$

【0039】これらのアドレスの具体例を図10に示す。図10は、所定のタイムユニットの先頭からはじまるMPEGビデオデータを示す。ここで、I、P、BはそれぞれIピクチャ、Pピクチャ、またはBピクチャを表し、また添え字の数字は、ピクチャの表示順序を示す。このタイムユニットには、I2で示すエントリーポイントのIピクチャが存在する。また、IピクチャI2の次のPピクチャはP5であり、IピクチャI2の次の次のPピクチャは、P8である。この時、上記式で演算されたI_start_offset_from_time_unit_address、I_end_offset_address、P1_end_offset_address、P2_end_offset_addressは、図に示す関係になる。

【0040】すなわち、I_end_offset_addressは、IピクチャI2の終了アドレスI_end_addressから、IピクチャI2の開始アドレスI_start_addressを減算した値とされている。P1_end_offset_addressは、PピクチャP5の終了アドレスP1_end_addressから、IピクチャI2の開始アドレスI_start_addressを減算した値とされている。さらにP2_end_offset_addressは、PピクチャP8の終了アドレスP2_end_addressから、IピクチャI2の開始アドレスI_start_addressを減算した値とされている。

【0041】I_start_offset_from_time_unit_address

は、エン트리ポイントのアドレス `Istart_address` から、タイムユニットのデータの先頭のアドレス `time_unit_address` を減算した値とされている。

【0042】なお、記録するトランスポートストリームの中に複数のプログラムが含まれる場合、エン트리ポイントの情報は、プログラム毎に区別して作成される。また、すべてのプログラムについて、エン트리ポイントデータを用意できない場合を考慮して、エン트리ポイントマップは、プログラム毎にエン트리ポイントデータが存在するかどうかを示す情報 (`parsed_program_flag`) を有する。

【0043】記録媒体に記録したトランスポートストリームを編集した場合、そのタイムユニットマップは変更(更新)される。次に、その方法を説明する。図11(A)は、図4に示すトランスポートストリームの先頭の2パケットと終わりの3パケットを消去する場合の例を示す。図11(B)は、このようにしてパケットが部分消去された後のトランスポートストリームを示す。図12は、図11(B)のトランスポートストリームのタイムユニットマップを示す。このようにタイムユニットの途中までのデータが消去された場合、最初のタイムユニットTU0の時間長 (`first_time_unit_size`) が変化するの、これを書き換えられる。図11(B)の場合は、タイムユニットTU0の時間長が、タイムユニットTU1の先頭のパケットPbのタイムスタンプと、消去後のタイムユニットTU0の先頭のパケットPaのタイムスタンプの差分値に変更される。また、図12に示すように、タイムユニットTU0の `delta_time_unit_address` が、タイムユニットTU1の先頭のパケットPbのアドレスA(TU1)と、消去後のタイムユニットTU0の先頭のパケットPaのアドレスCの差分値 $(A(TU1) - C)$ に更新される。タイムユニットマップが変更された場合は、それに関係するエン트리ポイントマップも変更される。

【0044】次に上述のタイムユニットマップのシンタックスの例を図13と図14に示す。図13と図14は、それぞれタイムユニットマップのヘッダ部 (`TimeUnitMapHeader()`) とデータ部 (`TimeUnitMapData()`) を表す。タイムユニットマップをファイルとして記録する時は、ヘッダ部とデータ部を1つのファイルにして記録しても良いし、別々のファイルとして記録しても良い。`TimeUnitMapHeader()` の `start_time`, `end_time` は、それぞれ、このタイムユニットマップの開始時刻と終了時刻を示し、例えば、あるトランスポートストリームを記録する時の記録開始時刻と記録終了時刻を示す。`first_time_unit_size` は、最初のタイムユニットの時間長を示す。`time_unit_size` は、第2番目以降のタイムユニットの時間長を示す。`number_of_time_unit_entries` は、トランスポートストリームの中のタイムユニットの数を示す。`TimeUnitMapData()` には、`number_of_time_unit_entries` で示される数の `delta_time_unit_address` (図7) が

書かれる。

【0045】また、上述のエン트리ポイントマップのシンタックスの第1の例を、図15乃至図17に示す。図15は、エン트리ポイントマップのヘッダ部 (`EntryPointMapHeader()`) を表し、図16はエン트리ポイントマップのデータ部 (`EntryPointMapData()`) を表す。図17は、さらに図16の `entry_point_data()` のシンタックスを表わしている。エン트리ポイントマップをファイルとして記録する時は、ヘッダ部とデータ部を1つのファイルにして記録しても良いし、別々のファイルとして記録しても良い。

【0046】図15の `EntryPointMapHeader()` の `number_of_programs` は、トランスポートストリームの中のプログラム数を示す。このシンタックスの第3行目から第6行目には、記録する各プログラムについて、エン트리ポイントマップテーブルが存在するかどうかを示す情報がある。第4行目の `program_number` は、プログラムを特定(識別)する情報であり、対応するプログラムのPMT (Program Map Table) に書かれている情報である。第5行目の `parsed_program_flag` は、そのプログラムのエン트리ポイントデータが存在するかどうかを示す。

【0047】第8行目から第10行目には、記録する各プログラムのPMTの情報が続く。`MPEG2_TS_program_map_section()` は、記録するトランスポートストリームの中から抽出した、MPEG2 systems規格で規定されているPMTである。ここで、`NUMBER_OF_ParsedPrograms` は、`parsed_program_flag` が「1」であるプログラムの数である。第8行目の `NUMBER_OF_ParsedPrograms` のループの中でデータが現れる順番は、第3行目の `number_of_programs` のループで `parsed_program_flag` が「1」である `program_number` が現れる順番である。

【0048】図16の `EntryPointMapData()` には、記録する各プログラムについてのエン트리ポイントのデータが記述される。1つのタイムユニットについてのエン트리ポイントのパラメータは、`entry_point_flag` と `entry_point_data()` である。1つのタイムユニットについての `entry_point_data()` の内容は、図17に示すように、`entry_point_time_stamp`, `I_start_offset_from_time_unit_address`, `I_end_offset_address`, `P1_end_offset_address`, `P2_end_offset_address` である。ここで、`entry_point_time_stamp` は、エン트리ポイントのトランスポートパケットのストリーム上の時刻、または、エン트리ポイントのIピクチャのPTS (Presentation Time Stamp) に基づいて計算される。PTSは、MPEG2システムズ規格のPESパケットのヘッダに付加されている情報である。

【0049】また、上述のエン트리ポイントマップのシンタックスの第2の例を、図18に示す。`EntryPointMapHeader()` と `entry_point_data()` の構成は、上述の第1の例における図15または図17に示す場合と同様で

ある。この図18と図16を比較して明らかなように、各プログラムについてのエントリーポイントのデータの並び方が、図16の第1の例とは異なる。

【0050】次に、以下に示す状態の第1の例と第2の例のそれぞれの場合のエントリーマップのデータの並びの例を示す。ここでは、図19に示すように、トランスポートストリームの中に3つのプログラム(program#1, program#2, program#3)が多重化されていて、タイムユニットTU_i(*i* = 0, 1, 2, 3)毎に、各プログラムのエントリーポイントがあるものとする。この場合、各パラメータは次のようになる。

【0051】

```
number_of_time_unit_entries = 4
number_of_programs = 3
program_number = 1 : parsed_program_flag = 1
program_number = 2 : parsed_program_flag = 1
program_number = 3 : parsed_program_flag = 1
NUMBER_OF_ParsedPrograms = 3
```

【0052】図20は、第1の例(図16の例)の場合のエントリーポイントマップを示す。この場合、プログラム毎にエントリーポイントデータのリストが別れた形になる。すなわち、program#1のEntryPointMapDataは、図20(A)に示すように、タイムユニットTU0乃至TU3のそれぞれに、entry_point_dataとして、entry_point_data#1-1乃至entry_point_data#1-4が存在するため、entry_point_flagはそれぞれ「1」とされる。

【0053】なお、entry_point_data#A-Bは、program_number=AのB番目のエントリーポイントについてのentry_point_data()を表わす。

【0054】program#2のEntryPointMapDataは、図20(B)に示すように、タイムユニットTU1, TU3には、entry_point_dataが存在しないため、そのentry_point_flagは「0」とされる。これに対して、タイムユニットTU0, TU2においては、それぞれentry_point_data#2-1, entry_point_data#2-2が存在するため、そのentry_point_flagは「1」とされる。

【0055】さらに、program#3のEntryPointMapDataのタイムユニットTU0, TU2には、entry_point_dataが存在しないため、entry_point_flagは「0」とされる。タイムユニットTU1, TU3には、entry_point_data#3-1, entry_point_data#3-2がそれぞれ存在するため、そのentry_point_flagは「1」とされている。

【0056】これらのentry_point_flagと、entry_point_dataが、EntryPointMapDataに記述される。

【0057】また、図21は、第2の例(図18の例)の場合のエントリーポイントマップを示す。

【0058】この場合、タイムユニット毎に各プログラムのエントリーポイントデータが時間順に並ぶ形になり、エントリーポイントデータのリストは1つの形となる。すなわち、タイムユニットTU0において、3つのプ

ログラムprogram#1乃至#3が記述され、それぞれについて、entry_point_flagと対応するentry_point_dataが記述される。この例では、program#3には、entry_point_dataが存在しないため、そのentry_point_flagは「0」とされ、program#1, #2については、entry_point_data#1-1, #2-1が存在するため、そのentry_point_flagは「1」とされている。

【0059】その他のタイムユニットTU1乃至TU3においても、program#1乃至#3それぞれについて、entry_point_flagと、entry_point_dataが記述される。

【0060】次に、図22を用いて、タイムユニットマップの別の例を説明する。図22(A), (B)は、前述の図3(A), (B)と同様の構成を示す。また、図22(C)は、図4と同様に、記録媒体上に記録されたソースパケットデータを示す。トランスポートパケットとソースパケットの関係は、図4に示すとおりである。図22(C)では、横軸は記録されたソースパケットの番号を示す。ソースパケット番号は、ソースパケットの記録された順番であり、ゼロから開始して、順番に1づつインクリメントする。タイムユニット毎のデータの先頭アドレスA(TU0), A(TU1), A(TU2)は、ソースパケット番号単位でカウントして、それぞれ0, 4, 7となる。

【0061】図22(D)は、タイムユニットマップ、すなわち記録されたトランスポートストリームのタイムユニット毎のデータの先頭アドレスのテーブルの例を示す。ここで、RSPN(Relative Source Packet Number) time_unit_startは、記録されたストリーム上のタイムユニットの先頭データのアドレスを示す。タイムユニットマップでは、記録されたトランスポートストリームの先頭からのタイムユニットの順にRSPN_time_unit_startがテーブル化され、図22(D)の場合、A(TU0), A(TU1), A(TU2)の順に値が並ぶ。

【0062】次に上述のタイムユニットマップのシンタックスの例を図23と図24に示す。図23と図24は、それぞれタイムユニットマップのヘッダ部(TimeUnitMapHeader())とデータ部(TimeUnitMapData())を表す。タイムユニットマップをファイルとして記録する時は、ヘッダ部とデータ部を1つのファイルにして記録しても良いし、別々のファイルとして記録しても良い。

【0063】図23のTimeUnitMapHeader()のoffset_SPNは、記録されたトランスポートストリームの一番目のソースパケットの番号を示す。本実施の形態では、RSPN_time_unit_startは、offset_SPNに対する相対的なソースパケット番号で表す。タイムユニットマップをはじめて作成する時の、offset_SPNの初期値は、ゼロである。次のoffset_timeは、このタイムユニットマップの開始時刻を示すものであり、一番目のタイムユニットの開始時刻を示す。

【0064】time_unit_sizeは、このタイムユニットマップのタイムユニットの時間長を示す。

【0065】number_of_time_unit_entriesは、記録されたトランスポートストリームの中のタイムユニットの数を示す。

【0066】図24のTimeUnitMapData()には、number_of_time_unit_entriesで示される数のRSPN_time_unit_start (図22 (D))が書かれる。

【0067】次に、記録されたトランスポートストリームを編集した場合の上述のタイムユニットマップの変更(更新)の方法を説明する。図25 (A)は、図22 (C)に示すトランスポートストリームの先頭の2パケット、すなわち先頭からCで示すアドレスのソースパケットまでを消去する場合の例を示す。図25 (B)は、このようにしてパケットが部分消去された後のトランスポートストリームを示す。

【0068】図26は、図25 (A)と(B)の場合のトランスポートストリームのタイムユニットマップを示す。このようにタイムユニットの先頭部分のデータが消去された場合、消去された部分のデータを含むタイムユニットを参照するRSPN_time_unit_startのデータが消去される。図26 (B)の場合、TU0のRSPN_time_unit_startのデータが消去される。これに伴って、number_of_time_unit_entriesが、消去したRSPN_time_unit_startのデータの数だけデクリメントされる。

【0069】また、上述のoffset_SPNとoffset_timeが変更される。図26 (A)の場合、offset_SPNは、ゼロであり、また、offset_timeは、TU0のタイムユニットの開始時刻のゼロがセットされている。図26 (B)の場合、offset_SPNが、新しく先頭になったソースパケットの元々のソースパケットの番号のC (=2)に変更される。また、offset_timeが新しく先頭になったタイムユニット、すなわち、この例ではTU1の元々の開始時刻の ΔT に変更される。なお、編集前後で、time_unit_sizeは、変更されない。

【0070】タイムユニットマップが変更された場合は、それに関係するエン트리ポイントマップも変更される。

【0071】次に、入力されたトランスポートストリームから、上述のテーブルを作成してトランスポートストリームとともに、記録媒体に記録する動画像記録装置1の構成例を図27に示す。

【0072】端子10から入力されるトランスポートストリームには、1つまたは複数のAVプログラムが多重化されている。端子22には、ユーザインタフェースによって選択されたAVプログラムのチャンネル(サービス名)が入力される。ここで選択されるチャンネル数は、1つでも複数でも良い。

【0073】PIDフィルタ11は、入力されたトランスポートストリームの中から、ストリーム解析部12により指定されたPID(Packet ID)のトランスポートパケットを取り出す。PIDは、トランスポートパケットのヘッダ

の固定位置にある13ビット長の符号であり、そのトランスポートパケットのペイロードにストアされているデータのタイプを表す。はじめにPIDフィルタ11は、PID=0x0000であるPAT(Program Association Table)のトランスポートパケットを取り出す。PATには、トランスポートストリームに多重化されている各プログラムのPMT(Program Map Table)のトランスポートパケットのPIDが書かれている。PIDフィルタ11から出力されるPATのトランスポートパケットは、ストリーム解析部12へ入力される。

【0074】カウンタ24は、記録するトランスポートストリームの先頭パケットから現在のパケットまでのパケット数を計数し、現在のパケットナンバーを、タイムユニットマップ作成部23とエン트리ポイントマップ作成部16へ出力する。

【0075】ストリーム解析部12は、PCR(Program Clock Reference)を伝送するトランスポートパケットからPCRを抽出して、PLL部13へ出力する。PCRを伝送するトランスポートパケットのPIDが複数ある場合は、どれか1つのPIDのパケットからPCRが抽出される。PLL部13は、入力されたPCRに同期して、27MHzの周波数のクロックを生成し、そのクロックをタイムスタンプ発生部14に出力する。

【0076】タイムスタンプ発生部14は、入力されたクロックをカウントし、そのカウント値に対応したタイムスタンプ(Arrival Time Stamp)を生成する。このタイムスタンプは、最初に記録するトランスポートパケットのタイムスタンプをゼロとすれば、そのトランスポートストリームの記録後の経過時間を表すことになる。このタイムスタンプは、ストリーム解析部12、タイムスタンプ付加部15、およびタイムユニットマップ作成部23へ出力される。

【0077】タイムスタンプ付加部15は、PIDフィルタ11から入力されたトランスポートパケットに、その到着時刻を示すタイムスタンプを含むヘッダーを付加し、ソースパケット(図4)をファイルシステム部17へ出力する。

【0078】タイムユニットマップ作成部23は、カウンタ24から入力されるパケットナンバーと、タイムスタンプ発生部14から入力されるタイムスタンプに基づいて、上述のタイムユニットマップを作成する。作成されたタイムユニットマップは、エン트리ポイントマップ作成部16とファイルシステム部17へ出力される。

【0079】ストリーム解析部12は、プログラム毎の次に示すプログラム情報をエン트리ポイントマップ作成部16へ出力する。(1)プログラムのprogram_number(2)プログラムのPMTのトランスポートパケットのPID(3)プログラムを構成するビデオのトランスポートパケットのPIDとstream_type(4)プログラムを構成するオーディオのトランスポートパケットのPIDとstream_ty

pe(5) プログラムのPCRのPID ここで、stream_type は、PMTに書いてある内容であり、ビデオの場合、MPEG 2/MPEG1などのストリームタイプを表し、またオーディオの場合、MPEG1/AC-3などのストリームタイプを表す。

【0080】ストリーム解析部12はまた、記録するストリームのエン트리ポイントデータを作成し、エン트리ポイントマップ作成部16へ入力する。エン트리ポイントデータの内容は、図17に示すものである。なお、エン트리ポイントのタイムスタンプをエン트리ポイントのPTSとする場合、PTSはストリーム解析部12が入力ストリームから取り出すので、タイムスタンプ発生部14により作成したタイムスタンプをストリーム解析部12へ入力する必要はない。

【0081】エン트리ポイントマップ作成部16は、エン트리ポイントデータをプログラム毎にテーブル化し、上述のエン트리ポイントマップを作成し、ファイルシステム部17へ出力する。

【0082】次に、その動作について説明する。PIDフィルタ11は、端子10からトランスポートストリームが入力されると、PID=0x0000であるPIDを含むトランスポート packets を抽出し、ストリーム解析部12に出力する。ストリーム解析部12は、この時、図28のフローチャートに示す処理を実行する。

【0083】ステップS11で、ストリーム解析部12は、PIDフィルタ11からPID=0x0000のトランスポート packets を受信すると、そのPATから、端子22を介して指令された各プログラムのPMTのトランスポート packets のPIDを取得する。

【0084】ステップS12で、ストリーム解析部12は、各プログラムのPMTのPIDをPIDフィルタ11にセットする。PIDフィルタ11は、これらPMTのPIDをもつトランスポート packets を取り出すと、それをストリーム解析部12へ出力する。

【0085】ステップS13で、ストリーム解析部12は、PIDフィルタ11からPMTのトランスポート packets を受信する。PMTには、そのプログラムを構成するビデオストリームやオーディオストリームをペイロードに持つトランスポート packets のPIDやPCR(Program Clock Reference)を伝送している packets のPIDが書かれている。ストリーム解析部12は、ユーザインタフェースによって選択された各プログラムを構成するビデオストリームやオーディオストリームをペイロードに持つトランスポート packets のPIDとPCRを伝送している packets のPIDをここで取得する。

【0086】ステップS14で、ストリーム解析部12は、ユーザインタフェースによって選択された各プログラムを構成するビデオストリームやオーディオストリームをペイロードに持つトランスポート packets のPIDとPCRを伝送している packets のPIDを、PIDフィルタ11

にセットする。

【0087】なお、あらかじめEPG(Electrical Program Guide)等を伝送するサービスインフォメーションの packets のPIDがわかっている場合は、これらのPIDもまた、PIDフィルタ11にセットされ、それらPIDの packets も、PIDフィルタ11から出力される。

【0088】このようにして、PIDフィルタ11により抽出されたトランスポート packets は、カウンタ24、ストリーム解析部12およびタイムスタンプ付加部15に供給される。カウンタ24は、記録するトランスポートストリームの先頭の packets から現在の packets までの packets 数を計数し、現在の packets ナンバを検知する。検知された現在の packets NO. は、タイムユニットマップ作成部23と、エン트리ポイントマップ作成部16へ供給される。

【0089】また、ストリーム解析部12は、入力されるトランスポート packets からPCRを抽出し、PLL部13へ供給する。PLL部13は、入力されたPCRに同期して、27MHzの周波数のクロックを生成し、タイムスタンプ発生部14に供給する。

【0090】タイムスタンプ発生部14は、入力されたクロックをカウントし、そのカウント値に対応するタイムスタンプを生成する。タイムスタンプ付加部15は、PIDフィルタ11から入力されたトランスポート packets に、その到着時刻を示す、タイムスタンプ発生部14が発生したタイムスタンプを付加し、ソース packets をファイルシステム部17に供給する。

【0091】タイムユニットマップ作成部23は、カウンタ24から入力される packets ナンバと、タイムスタンプ発生部14から入力されるタイムスタンプに基づいて、図7に示したようなタイムユニット毎のtime_unit_addressと、delta_time_unit_addressとを対応させたタイムユニットマップを作成し、エン트리ポイントマップ作成部16と、ファイルシステム部17へ供給する。または、タイムユニットマップ作成部23は、図22(D)に示したようなタイムユニット毎のRSPN_time_unit_startを対応づけたタイムユニットマップを作成し、エン트리ポイントマップ作成部16と、ファイルシステム部17へ供給する。

【0092】ストリーム解析部12はまた、プログラム毎の上述したプログラム情報を、エン트리ポイントマップ作成部16へ供給する。

【0093】このため、ストリーム解析部12は、図29と図30に示すような、エン트리ポイントの解析処理を実行する。

【0094】ステップS31でストリーム解析部12は、記録するプログラムのビデオのPIDと、そのstream_typeをPIDフィルタ11にセットする。これにより、PIDフィルタ11から、指定したビデオの packets が、ストリーム解析部12に供給される。

【0095】ステップS32でストリーム解析部12は、ビデオパケットのポインタvppを初期化し、vpp=0とする。ポインタvppは、現在処理している上記PIDのビデオパケットの順番を表す。

【0096】ステップS33でストリーム解析部12は、ビデオパケットのポインタvppをインクリメントする（例えば、1だけ増加する）。

【0097】ステップS34で、ストリーム解析部12は、ペイロードの中のストリームに、MPEGビデオのsequence_header_code(32ビット長で"0x000001B3"の符号)が含まれているか否かを調べる。sequence_header_codeが含まれていない時は、処理はステップS33へ戻る。

【0098】ステップS34で、ペイロードにsequence_header_codeが含まれていると判定された時は、ステップS35へ進み、ストリーム解析部12は、sequence_header_codeを含むパケット（最初のIピクチャのパケット）のアドレスをI_start_addressとする（図10）。

【0099】ステップS36でストリーム解析部12は、ビデオパケットのポインタvppをインクリメントする。

【0100】ステップS37で、ストリーム解析部12は、上記Iピクチャのデータが終了したか否かを調べる。Iピクチャのデータがまだ終了していない場合、処理はステップS36へ戻る。Iピクチャのデータが終了した場合、処理はステップS38へ進む。

【0101】ステップS38で、ストリーム解析部12は、Iピクチャが終了するパケットのアドレスをI_end_addressとする（図10）。以上により、最初のIピクチャのアドレスが決定されたことになる。

【0102】ストリーム解析部12は、ステップS39で（ビデオポインタvppはインクリメントしないで）、次のビデオパケットがシーケンスヘッダコードを含んでいるか否かを調べる。パケットがシーケンスヘッダコードを含んでいる場合、処理はステップS47へ進む。パケットがシーケンスヘッダコードを含んでいない場合、処理はステップS40へ進む。

【0103】ストリーム解析部12は、ステップS40でビデオパケットのポインタvppをインクリメントする。

【0104】ストリーム解析部12は、ステップS41で、PピクチャまたはIピクチャが終了したかどうかを調べる。PピクチャまたはIピクチャが終了していない場合、処理はステップS39へ戻る。PピクチャまたはIピクチャが終了している場合、処理はステップS42へ進む。

【0105】ストリーム解析部12は、ステップS42で、PまたはIピクチャが終了するパケットのアドレスをP1_end_addressとする（図10）。以上により、Iピクチャの次の最初のPピクチャまたはIピクチャのアドレス

が決定されたことになる。

【0106】ストリーム解析部12は、ステップS43で（ビデオポインタvppはインクリメントしないで）、次のビデオパケットがシーケンスヘッダコードを含んでいるか否かを調べる。ビデオパケットがシーケンスヘッダコードを含んでいる場合、処理はステップS47へ進む。ビデオパケットがシーケンスヘッダコードを含んでいない場合、処理はステップS44へ進む。

【0107】ストリーム解析部12は、ステップS44でビデオパケットのポインタvppをインクリメントする。

【0108】ストリーム解析部12は、ステップS45でPピクチャまたはIピクチャが終了したかどうかを調べる。PピクチャまたはIピクチャが終了していない場合、処理はステップS43へ戻る。PピクチャまたはIピクチャが終了している場合、処理はステップS46へ進む。

【0109】ストリーム解析部12は、ステップS46で、PまたはIピクチャが終了するパケットのアドレスを、P2_end_addressとする（図10）。以上により、Iピクチャの次の次のPピクチャまたはIピクチャのアドレスが決定されたことになる。

【0110】ストリーム解析部12は、ステップS47でI_start_address, I_end_address, P1_end_address, P2_end_addressのアドレスを、エントリーポイントマップ作成部16へ出力する。なお、この時、P1_end_addressとP2_end_addressの少くとも一方は存在しない場合もある。

【0111】ストリーム解析部12は、ステップS48で、現在のパケットが最後の入力パケットであるかどうかを判定する。現在のパケットが最後のパケットでない場合、処理はステップS33へ戻る。現在のパケットが最後のパケットである場合、処理は終了される。

【0112】以上のビデオストリームの解析は、記録するトランスポートストリームの中に複数のプログラムがある場合は、それぞれのプログラムのビデオパケットに対して行なわれる。

【0113】ストリーム解析部12は、以上のようにしてエントリーポイントデータを生成すると、これをエントリーポイントマップ作成部16に供給する。エントリーポイントマップ作成部16は、ストリーム解析部12より供給されたエントリーポイントデータを、プログラム毎にテーブル化し、図9に示すようなエントリーポイントマップを作成し、ファイルシステム部17に供給する。

【0114】以上のようにして、ファイルシステム部17には、タイムスタンプ付加部15によりタイムスタンプが付加されたトランスポートストリームと、その特徴点を表わす特徴点データとしてのタイムユニットマップと、エントリーポイントマップが、タイムユニットマップ作成部23とエントリーポイントマップ作成部16か

らそれぞれ供給される。ファイルシステム部17は、トランスポートストリームと、それに対応する特徴点データをファイル化する。

【0115】図31は、このファイル構造の例を表わしている。この例においては、トランスポートストリームファイルの中に、3個のプログラムが多重化されている。同図に示すように、エントリーポイントマップは、タイムユニットマップに従属する構成とされている。そして、各エントリーポイントマップは、プログラム毎にそれぞれ次のデータを有する。

- (1) プログラムのprogram_number
- (2) プログラムのPMTのトランスポートパケットのPID
- (3) プログラムを構成するビデオのトランスポートパケットのPIDとstream_type
- (4) プログラムを構成するオーディオのトランスポートパケットのPIDとstream_type
- (5) プログラムのPCRのPID (6) エントリーポイントのリスト

【0116】ファイルシステム部17により生成されたファイルは、誤り訂正部18に供給され、誤り訂正符号が付加された後、変調部19に供給され、所定的方式で変調される。変調部19より出力された信号は、書き込み部20に供給され、記録媒体21に書き込まれる。

【0117】以上のようにして、トランスポートストリームとその特徴点データが、記録媒体21に記録される。

【0118】以上においては、タイムユニットマップとエントリーポイントマップを、トランスポートストリームから作成するようにしたが、例えば、動画像記録装置自身が、トランスポートストリームを多重化し、生成するような場合、その多重化動作時に、タイムユニットマップとエントリーポイントマップを、作成するようにすることもできる。図32は、この場合の構成例を表わしている。

【0119】すなわち、図32の例においては、多重化部40に複数(n個)のプログラムの、ビデオとオーディオのエレメンタリーストリーム#1乃至#nが入力されている。システムタイムクロック部42は、27MHzの周波数のシステムタイムクロックをカウントし、タイムスタンプを生成し、コントローラ41とタイムユニットマップ作成部43に出力している。コントローラ41は、多重化部40に入力された各エレメンタリーストリームを解析し、多重化部40が、MPEG2システム規格のT-STD(Transport Stream System Target Decoder)を満たして、トランスポートストリームを多重化するように、多重化部40を制御する。

【0120】コントローラ41は、多重化部40から出力される、トランスポートパケットの数を示すパケットナンバーを、タイムユニットマップ作成部43とエントリーポイントマップ作成部44に出力する。タイムユニ

ットマップ作成部43は、コントローラ41より入力されるパケットナンバと、システムタイムクロック42より入力されるタイムスタンプに基づいて、タイムユニットマップを生成する。

【0121】コントローラ41はまた、プログラム情報とエントリーポイントデータとを、エントリーポイントマップ作成部44に出力する。エントリーポイントマップ作成部44は、コントローラ41より供給される、パケットナンバ、プログラム情報、およびエントリーポイントデータ、並びにタイムユニットマップ作成部43より供給されるタイムユニットマップに基づいて、エントリーポイントマップを生成する。

【0122】多重化部40より出力されたトランスポートストリーム、タイムユニットマップ作成部43により作成されたタイムユニットマップ、およびエントリーポイントマップ作成部44により作成されたエントリーポイントマップは、それぞれ、図27に示したファイルシステム部17に供給される。ファイルシステム部17乃至記録媒体21までの構成は、図27に示した場合と同様である。

【0123】この図32に示すような構成の動画像記録装置1においては、コントローラ41が、多重化部40により多重化されるエレメンタリーストリームから、プログラム情報と、エントリーポイントデータを生成し、エントリーポイントマップ作成部44に出力する。また、コントローラ41は、システムタイムクロック42より入力されるタイムスタンプに対応するパケットナンバを、タイムユニットマップ作成部43とエントリーポイントマップ作成部44に出力する。

【0124】タイムユニットマップ作成部43は、コントローラ41から入力されるパケットナンバと、システムタイムクロック42より入力されるタイムスタンプに基づいて、タイムユニットマップを作成する。同様に、エントリーポイントマップ作成部44は、コントローラ41より入力されるパケットナンバ、プログラム情報、およびエントリーポイントデータ、並びにタイムユニットマップ作成部43より入力されるタイムユニットマップに基づいて、エントリーポイントマップを作成する。

【0125】そして、作成されたトランスポートストリーム、タイムユニットマップおよびエントリーポイントマップは、図27に示した場合と同様に、ファイルシステム部17によりファイル化され、誤り訂正部18により誤り訂正分が付加される。そして変調部19によりさらに変調された後、書き込み部20により、記録媒体21に記録される。

【0126】次に、以上のようにして、トランスポートストリームファイルと、そのストリームの特徴点データが記録された記録媒体21を再生する動画像再生装置について説明する。図33は、このような動画像再生装置51の構成例を表わしている。読み出し部61は、記録

媒体21に記録されているデータを読み出し、復調部62に出力する。復調部62は、読み出し部61より入力されたデータを復調して、誤り訂正部63に出力する。誤り訂正部63は、復調部62より入力されたデータの誤りを訂正し、ファイルシステム部64に供給する。

【0127】ファイルシステム部64は、誤り訂正部63より入力されたデータを、トランスポートストリームファイルと、特徴点データとに分離し、ストリームファイルをバッファ65に供給するとともに、特徴点データを再生制御部71に出力する。再生制御部71は、端子73からユーザインタフェースを介して、ユーザより入力された指令に対応して、読み出し部61、デマルチプレクサ69、およびAVデコーダ70を制御する。

【0128】バッファ65は、ストリームファイルからarrival_time_stampに格納されている値を初期値として、カウンタ68に供給し、セットさせる。カウンタ68は、システムタイムクロック部67が発生する27MHzの周波数のクロックを、バッファ65からの初期値を基準としてカウントし、カウント値を比較部66に供給する。

【0129】比較部66は、カウンタ68より供給されたカウンタの値と、バッファ65より供給されたトランスポートパケットに含まれているarrival_time_stampの値を比較し、両者の値が一致したとき、そのトランスポートパケットをデマルチプレクサ69に出力する。

【0130】デマルチプレクサ69は、比較部66より入力されたトランスポートストリームファイルから、再生制御部71からの指令に対応するチャンネルのビデオデータと、オーディオデータとを抽出し、AVデコーダ70に出力する。AVデコーダ70は、デマルチプレクサ69より入力された、ビデオデータとオーディオデータをデコードし、端子72から出力する。

【0131】次に、その動作について説明する。記録媒体21には、図27（または図32）の動画像記録装置1で記録したトランスポートストリームファイルと、そのストリームの特徴点データが記録されている。トランスポートストリームファイルには、1つまたは複数のプログラムが多重化されている。

【0132】はじめに再生制御部71は、読み出し部61に対して、ストリームの特徴点データを読み出すように指示する。このとき、読み出し部61は、記録媒体21からストリームの特徴点データを読み出し、復調部62に出力する。復調部62は、入力されたデータを復調し、誤り訂正部63に出力する。誤り訂正部63は、入力されたデータの誤りを訂正し、ファイルシステム部64に供給する。ファイルシステム部64は、入力されたストリーム特徴点データを再生制御部71に出力する。

【0133】端子73からは、ユーザインタフェースによって再生を指定されたプログラム番号が入力され、それが再生制御部71へ入力される。再生制御部71は、

そのプログラムのPMTのトランスポートパケットのPID、プログラムを構成するビデオのトランスポートパケットのPIDとstream_type、プログラムを構成するオーディオのトランスポートパケットのPIDとstream_type、並びにPCRのPIDを、特徴点データから読み出し、デマルチプレクサ69とAVデコーダ70へ出力する。

【0134】さらに、再生制御部71は、読み出し部61に対して、トランスポートストリームファイルを読み出すように指示する。この指令に対応して、読み出し部61は、記録媒体21からトランスポートストリームファイルを読み出す。このデータは、上述した場合と同様に復調部62、誤り訂正部63、ファイルシステム部64の処理を経て、バッファ65へ入力される。

【0135】バッファ65は、入力されたトランスポートストリームファイルから、arrival_time_stampに記憶されている値を読み出し、カウンタ68に初期値として供給し、セットさせる。カウンタ68は、この初期値を基準として、システムタイムクロック部67が発生するクロックをカウントし、そのカウント値を比較部66に供給する。比較部66は、バッファ65より供給されるトランスポートストリームファイルから、arrival_time_stampの値を読み出し、カウンタ68から供給されるカウンタ値と比較する。両者の値が一致するタイミングで、比較部66は、トランスポートストリームファイルを、デマルチプレクサ69に出力する。

【0136】デマルチプレクサ69は、ユーザインタフェースにより指定されたプログラムを構成するビデオとオーディオのトランスポートパケットを、入力されたトランスポートストリームから分離し、それをAVデコーダ70へ入力する。AVデコーダ70は、ビデオストリームとオーディオストリームを復号し、再生ビデオ信号と再生オーディオ信号として端子72から出力する。

【0137】ユーザインタフェースによってランダムアクセス再生が指示された場合、再生制御部71は、内部に記憶されているストリームの特徴点データの内容に基づいて、記録媒体21からのデータの読み出し位置を決定し、ランダムアクセス制御情報を読み出し部61へ入力する。例えば、ユーザによって選択されたプログラムを所定の時刻から途中再生する場合、再生制御部71は、タイムユニットマップに基づいて、指定された時刻に対応するトランスポートストリームのアドレスを計算し、そのアドレスからデータを読み出すように読み出し部61へ指示する。以下に、その手順を説明する。

【0138】はじめに、図7に示したタイムユニットマップの場合を説明する。ゼロ番目のタイムユニットTU0の先頭データの時刻をstart_timeとすれば、N番目(N>0)のタイムユニットの先頭データの時刻は、(start_time+first_time_unit_size+(N-1)*time_unit_size)となる。ユーザから指定された時刻よりもタイムユニットの先頭データの時刻が大きくなるタイムユニットの番号

がわかったら、その番号のタイムユニットからデータを読み出せば良いことがわかる。

【0139】この場合、記録されたストリーム上の0番目のタイムユニットの先頭データのアドレスを0とすれば、N番目のタイムユニットの先頭データのアドレス $time_unit_address(N)$ は、次の様に計算できる。

$$time_unit_address(N) = \sum_{i=0}^{N-1} \Delta time_unit_address(i)$$

【0141】次に、図22(D)に示したタイムユニットマップの場合を説明する。

【0142】この場合、N番目($N \geq 0$)のタイムユニットの先頭データの時刻は、($offset_time + N * time_unit_size$)となる。ユーザから指定された時刻よりもタイムユニットの先頭データの時刻が大きくなるタイムユニットの番号がわかったら、その番号のタイムユニットからデータを読み出せば良いことがわかる。N番目のタイムユニットの先頭データのソースパケット番号は、($RSPN_time_unit_start(N) - offset_SPN$)となる。ここで、 $RSPN_time_unit_start(N)$ は、N番目のタイムユニットについての $RSPN_time_unit_start$ の値である。

【0143】また、ユーザによって選択されたプログラムに対応するエン트리ポイントマップのデータが存在する場合、再生制御部71は、エン트리ポイントデータに基づいて、特殊再生を制御できる。例えば、高速再生の場合、再生制御部71は、エン트리ポイント毎のアドレスのストリームデータを順次連続して読み出すように読み出し部61へ指示する。

【0144】図34は、この場合の再生制御部71の動作を表わしている。再生制御部71は、ステップS61で、ユーザからの指令に対応して、内蔵するメモリに、再生するプログラムの $program_number$ をセットする。

【0145】再生制御部71は、ステップS62で、 $parsed_program_flag$ から、そのプログラムのエン트리ポイントデータが存在するか否かを調べる。存在する($parsed_program_flag = 1$ である)場合は、ステップS63へ進む。エン트리ポイントデータが存在しない場合は、エン트리ポイントマップを使用したデータアクセスはできないので、処理は終了される。

【0146】再生制御部71は、ステップS63で、ユーザにより指定された時刻から読み出し開始するタイムユニットの番号 TN を上述のようにして計算する。すなわち、図7に示したタイムユニットマップの場合、 $start_time + first_time_unit_size + (N-1) * time_unit_size$ の値(タイムユニットの先頭の時刻)が、指定された時刻よりも大きくなるタイムユニットの番号 TN が計算される。または、図22(D)に示したタイムユニットマップの場合、($offset_time + N * time_unit_size$)が、指定された時刻よりも大きくなるタイムユニットの番号 TN が計算される。

【0147】再生制御部71は、ステップS64で、 TN

$e_unit_address(N)$ は、次の様に計算できる。

【0140】

【数1】

番号のタイムユニットに、そのプログラムのエン트리ポイントが存在するか否かを、 $entry_point_flag$ から調べる。エン트리ポイントが存在する($entry_point_flag = 1$ である)場合は、ステップS65へ進む、存在しない場合は、ステップS67へ進む。

【0148】再生制御部71は、エン트리ポイントが存在する場合、ステップS65で、 $entry_point_data()$ からエン트리ポイントのストリームデータを読み出すアドレスを計算する。ストリームデータの読み出し開始アドレスは、 $I_start_address$ であり、読み出し終了アドレスは、 $I_end_address$ 、 $P1_end_address$ 、または $P2_end_address$ である。

【0149】再生制御部71は、ステップS66で、ステップS65で計算したアドレスに基づいて、エン트리ポイントのストリームデータを読み出すように読み出し部61に指示する。読み出し部61はこの指示に対応して読み出し動作を実行する。

【0150】再生制御部71は、ステップS67で、番号 TN をインクリメントする。再生制御部67は、ステップS68で、処理の終了が指令されたか否かを判定し、処理の終了が指令されていない場合は、ステップS64へ戻り、そうでない場合は処理を終了する。

【0151】読み出し部61は、指定されたランダムアクセスポイントからデータを読み出す。読み出されたデータは、復調部62、誤り訂正部63、ファイルシステム部64バッファ65、比較部66の処理を経て、デマルチプレクサ69へ入力され、AVデコーダ70で復号され、出力される。

【0152】このステップS63の計算処理の詳細について、図7に示したタイムユニットマップの場合を、図35と図36のフローチャートを参照してさらに説明する。ステップS81において、再生制御部71に、端子73から $program_number$ と、再生開始時刻 Tst が入力されると、ステップS82において、再生制御部71は、ステップS81で入力された再生開始時刻 Tst が、特徴点データに含まれる、トランスポートストリームの開始時刻 $start_time$ (図3(B))と等しいか否かを判定する。再生開始時刻 Tst が開始時刻 $start_time$ と等しい場合には、ステップS86に進み、再生制御部71は、タイムユニットの番号を表わす変数 N に0を設定し、そのタイムユニット(0番目のタイムユニット)の $time_unit_address(N)$ に0を設定する。

【0153】これに対して、ステップS82において、再生開始時刻Tstが開始時刻starttimeと等しくないと判定された場合、ステップS83に進み、再生制御部71は、タイムユニットマップのヘッダ部を読み込み、ステップS84において、次の不等式を満たす最小の値Nを計算する。

【0154】 $Tst \Delta start_time + first_time_unit_size + (N-1) \times time_unit_size$ ステップS85において、再生制御部71は、タイムユニットマップのデータに基づいて、数1で示す式に従って、 $time_unit_address(N)$ を演算する。

【0155】N番目のタイムユニットの先頭データの時刻 $time_unit_address(N)$ が求められた時、ステップS87において、再生制御部71は、N番目のタイムユニットのアドレス $time_unit_address(N)$ からのデータ読み出しを、読み出し部61に指示する。

【0156】読み出し部61は、再生制御部71からの指令に対応して、ステップS88において、アドレス $time_unit_address(N)$ からのトランスポートストリームを記録媒体21から読み出す。読み出されたデータは、復調部62、誤り訂正部63、ファイルシステム部64、バッファ65、比較部66を介して、デマルチプレクサ69に供給される。

【0157】ステップS89において、再生制御部71は、デマルチプレクサ69に対して、ユーザより再生が指示された、プログラムのprogram_numberを出力する。デマルチプレクサ69は、ステップS90において、再生制御部71より指示された、program_numberのプログラムのトランスポートパケットを分離し、AVデコーダ70に出力する。ステップS91において、AVデコーダ70は、デマルチプレクサ69より入力されたデータをデコードし、端子72から出力する。

【0158】また、図34のフローチャートのステップS63の計算処理の詳細について、図22(D)に示したタイムユニットマップの場合を、図37のフローチャートを参照してさらに説明する。

【0159】図37のフローチャートは、図35と図36のフローチャートのステップS82、S84、S85を、それぞれステップS102、S104、S105へ変更したものである。これら以外は、図37のフローチャートは、図35と図36のフローチャートと同じである。以下では、ステップS102、S104、S105の各ステップについてのみ説明する。

【0160】ステップS102では、再生開始時刻Tstがタイムユニットマップの開始時刻offset_timeと比較される。

【0161】ステップS104においては、次の不等式を満たす最小の値Nが計算される。 $Tst < offset_time + N \times time_unit_size$

【0162】ステップS85においては、再生制御部7

1が、タイムユニットマップのデータに基づいて、次に示す式に従って、 $time_unit_address(N)$ を演算する。 $time_unit_address(N) = RSPN_time_unit_start(N) - offset_SPN$

【0163】上述した一連の処理は、ハードウェアにより実行させることもできるが、ソフトウェアにより実行させることもできる。一連の処理をソフトウェアにより実行させる場合には、そのソフトウェアを構成するプログラムが、専用のハードウェアとしての動画像記録再生装置に組み込まれているコンピュータ、または、各種のプログラムをインストールすることで、各種の機能を実行することが可能な、例えば汎用のパーソナルコンピュータなどにインストールされる。

【0164】次に、図38を参照して、上述した一連の処理を実行するプログラムをコンピュータにインストールし、コンピュータによって実行可能な状態とするために用いられる記録媒体について、そのコンピュータが汎用のパーソナルコンピュータである場合を例として説明する。

【0165】プログラムは、図38(A)に示すように、コンピュータ301に内蔵されている記録媒体としてのハードディスク302や半導体メモリ303に予めインストールした状態でユーザに提供することができる。

【0166】あるいはまた、プログラムは、図38(B)に示すように、フロッピーディスク311、CD-ROM(Compact Disk-Read Only Disk)312、MO(Magnetic-Optical)ディスク313、DVD(Digital Versatile Disk)314、磁気ディスク315、半導体メモリ316などの記録媒体に、一時的あるいは永続的に格納し、パッケージソフトウェアとして提供することができる。

【0167】さらに、プログラムは、図38(C)に示すように、ダウンロードサイト321から、デジタル衛星放送用の人工衛星322を介して、コンピュータ323に無線で転送したり、ローカルエリアネットワーク、インターネットといったネットワーク131を介して、コンピュータ323に有線で転送し、コンピュータ323において、内蔵するハードディスクなどに格納させることができる。

【0168】本明細書における記録媒体とは、これら全ての記録媒体を含む広義の概念を意味するものである。

【0169】また、本明細書において、記録媒体により提供されるプログラムを記述するステップは、記載された順序に沿って時系列的に行われる処理はもちろん、必ずしも時系列的に処理されなくとも、並列的あるいは個別に実行される処理をも含むものである。

【0170】なお、本明細書において、システムとは、複数の装置により構成される装置全体を表すものである。

【0171】このように、1つまたは複数のトランスポ

ートストリームが記録された記録媒体にランダムアクセスして再生する場合、1ピクチャやオーディオフレームの開始位置を効率よくサーチすることができるので、ユーザ入力に対して応答の速いランダムアクセス再生をすることができる。

【0172】

【発明の効果】以上の如く、本発明のデータ処理装置、データ処理方法、および第1の記録媒体のプログラムによれば、符号化ストリームを所定のタイムユニット毎に区分し、区分された符号化ストリームのタイムユニット毎のデータのアドレスを示すタイムユニットマップを作成するようにしたので、応答の速いランダムアクセスが可能となる。

【0173】本発明のデータ再生装置、データ再生方法、および第2の記録媒体のプログラムによれば、記録媒体から再生されたタイムユニットマップに基づいて、記録媒体に記録されている符号化ストリームを、任意の位置から再生するようにしたので、また、本発明の記録媒体によれば、符号化ストリームとともに、符号化ストリームのタイムユニット毎のデータのアドレスを示すタイムユニットマップをファイル化して記録するようにしたので、迅速なランダム生成が可能となる。

【図面の簡単な説明】

【図1】従来のトランスポートストリームのバケットを説明する図である。

【図2】従来の記録されるトランスポートストリームを説明する図である。

【図3】本発明のトランスポートストリームを説明する図である。

【図4】本発明の記録されるトランスポートストリームを説明する図である。

【図5】Source Packetのシンタックスを示す図である。

【図6】TP_extra_headerのシンタックスを示す図である。

【図7】タイムユニットマップの例を示す図である。

【図8】タイムユニット毎のオフセットアドレスを説明する図である。

【図9】エントリーポイントマップの例を示す図である。

【図10】エントリーポイントデータを説明する図である。

【図11】データの消去を説明する図である。

【図12】データを消去した時のタイムユニットマップの例を示す図である。

【図13】TimeUnitMapHeader()のシンタックスを示す図である。

【図14】TimeUnitMapData()のシンタックスを示す図である。

【図15】EntryPointMapHeader()のシンタックスを示す図である。

す図である。

【図16】EntryPointMapData()のシンタックスを示す図である。

【図17】entry_point_data()のシンタックスを示す図である。

【図18】EntryPointMapData()のシンタックスを示す図である。

【図19】トランスポートストリームファイルのエントリーポイントを説明する図である。

【図20】EntryPointMapDataの例を示す図である。

【図21】EntryPointMapDataの例を示す図である。

【図22】本発明のトランスポートストリームを説明する図である。

【図23】TimeUnitMapHeader()のシンタックスを示す図である。

【図24】TimeUnitMapData()のシンタックスを示す図である。

【図25】データの消去を説明する図である。

【図26】データを消去したときのタイムユニットマップの例を示す図である。

【図27】本発明を適用した動画像記録装置の構成例を示すブロック図である。

【図28】図27の動画像記録装置の動作を説明するフローチャートである。

【図29】図27の動画像記録装置の動作を説明するフローチャートである。

【図30】図27の動画像記録装置の動作を説明するフローチャートである。

【図31】トランスポートストリームファイルのタイムユニットマップとエントリーポイントマップの関係を説明する図である。

【図32】本発明を適用した動画像記録装置の他の構成例を示すブロック図である。

【図33】本発明を適用した動画像再生装置の構成例を示すブロック図である。

【図34】図33の動画像再生装置の動作を説明するフローチャートである。

【図35】図33の動画像再生装置の動作を説明するフローチャートである。

【図36】図33の動画像再生装置の動作を説明するフローチャートである。

【図37】図33の動画像再生装置の動作を説明するフローチャートである。

【図38】記録媒体を説明する図である。

【符号の説明】

1 動画像記録装置, 11 PIDフィルタ, 12 ストリーム解析部, 14 タイムスタンプ発生部, 15 タイムスタンプ付加部, 16 エントリーポイントマップ作成部, 17 ファイルシステム部, 21 記録媒体, 23 タイムユニットマップ作成部, 24

カウンタ, 40 多重化部, 41 コントローラ,
42 システムタイムクロック部, 43 タイムユ
ニットマップ作成部, 44 エントリーポイントマップ

作成部, 61 読み出し部, 69 デマルチプレク
サ, 70 AVデコーダ, 71 再生制御部

フロントページの続き

(72)発明者 中村 政信
東京都品川区北品川6丁目7番35号 ソニ
ー株式会社内

Fターム(参考) 5C053 FA14 FA20 FA23 FA24 FA27
GB06 GB08 GB38 HA29 KA19
5C059 KK39 MA00 PP05 PP06 PP07
RB02 RB09 RB10 RB16 RC04
SS11 SS13 SS20 SS26 UA36
UA39
5D044 AB05 AB07 DE02 DE03 DE38
DE40 DE53 DE96 EF02 FG19
GK08 GK12